ATTITUDE PROPAGATION

Samuele Vincenzi

1 Attitude propagation tool

Attitude state selection

The attitude dynamics of a spacecraft are most reliably and efficiently propagated using a quaternion-based representation for orientation. Quaternions offer several significant advantages:

- **Compactness**: Quaternions describe a rotation with only four parameters, making them more efficient than a 3x3 rotation matrix which requires nine elements.
- **No Singularities**: Unlike Euler angles, quaternions do not suffer from gimbal lock, ensuring a continuous and unambiguous representation of all possible orientations.

Consequently, the rotational state vector is formed as follows:

$$\mathbf{x}_{\text{rot}} = \begin{bmatrix} \mathbf{q} \\ \boldsymbol{\omega} \end{bmatrix}_{7\times 1}$$

Rotational dynamics

The time evolution of the angular rates is given by the Euler's equations in body coordinates with body torque $\boldsymbol{M}$:

$$I\,\dot{\boldsymbol{\omega}} \;=\; \boldsymbol{M} \;-\; \boldsymbol{\omega}\times\big(I\,\boldsymbol{\omega}\big), \qquad \dot{\boldsymbol{\omega}} \;=\; I^{-1}\Big(\boldsymbol{M} \;-\; \boldsymbol{\omega}\times\big(I\,\boldsymbol{\omega}\big)\Big).$$

Quaternion kinematics

Let $q = \begin{bmatrix} x & y & z & w \end{bmatrix}^{\mathsf{T}}$ and define the pure quaternion $\tilde{\omega} = \begin{bmatrix} \omega_x & \omega_y & \omega_z & 0 \end{bmatrix}^{\mathsf{T}}$. The kinematics $\dot{q} = \frac{1}{2} q \otimes \tilde{\omega}$ are the Hamilton (right-driven) quaternion differential equation consistent with body-frame rates and a passive inertial-to-body attitude Using the Hamilton product (right-driven by body rates), the quaternion kinematics are

$$\dot{q} \;=\; \tfrac{1}{2}\, L(q)\, \tilde{\omega},$$

where the 4×4 left-product matrix $L(q)$ implements $q \otimes r = L(q)\, r$ for any quaternion r:

$$L(q) \;=\; \begin{bmatrix} w & -z & y & x \\ z & w & -x & y \\ -y & x & w & z \\ -x & -y & -z & w \end{bmatrix}.$$

Equivalently, with $q = (\mathbf{v}, w)$ and $\mathbf{v} = \begin{bmatrix} x & y & z \end{bmatrix}^{\mathsf{T}}$,

$$\dot{\mathbf{v}} \;=\; \tfrac{1}{2}(w\,\boldsymbol{\omega} \;+\; \mathbf{v}\times\boldsymbol{\omega}), \qquad \dot{w} \;=\; -\tfrac{1}{2}\,\mathbf{v}\cdot\boldsymbol{\omega}.$$

The unit-norm constraint is enforced numerically:

$$\|q\| \;=\; 1.$$

Numerical integration in Python

The explicit Runge-Kutta method `RK45` is selected due to its strong balance of accuracy and computational efficiency. This adaptive, explicit method is well-suited for non-stiff problems such as spacecraft rotational dynamics, providing accurate integration without the additional computational overhead associated with implicit solvers. Since the rotational dynamics system does not exhibit stiffness, implicit solvers, which are generally used to handle stiff systems, are unnecessary, enabling faster integration with reliable accuracy.

The propagator function is designed to accept an input attitude representation in one of the following formats: **Direction Cosine Matrix (DCM)**, **quaternion**, or **Euler angles**. It returns the time evolution of quaternion, Euler angles, DCM, and angular rates (see section 2). Internally, these conversions are handled by a dedicated attitude conversion module, ensuring seamless interoperability between representations and flexibility for various applications.

2 Attitude conversion tool

This tool facilitates the conversion between three widely used representations of spatial orientation in aerospace: *quaternions*, *direction cosine matrices* (DCM), and *Euler angles*. Each representation has its advantages and drawbacks, and this tool automates the transformation among them for easier attitude management.

Quaternion to Direction Cosine Matrix (DCM)

A *quaternion* representing orientation is expressed as

$$q = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 \end{bmatrix},$$

where q_4 is typically the scalar component, and q_1, q_2, q_3 are vector components. The associated *Direction Cosine Matrix* C is a 3×3 orthogonal rotation matrix given by:

$$C = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix}.$$

This matrix satisfies orthogonality $CC^T = I$ and has determinant $+1$, representing a proper rotation. The formula derives from quaternion algebra and maps the quaternion rotation to a 3D rotation in Cartesian coordinates.

Direction Cosine Matrix (DCM) to Quaternion

Reconstructing a quaternion from a known DCM matrix C involves computing the quaternion components from C's trace and elements. The trace of C is

$$\text{trace}(C) = C_{11} + C_{22} + C_{33}.$$

If $\text{trace}(C) > 0$, the scalar component is

$$q_4 = \frac{1}{2}\sqrt{1 + \text{trace}(C)},$$

and the vector components are extracted from the anti-symmetric parts of C:

$$q_1 = \frac{C_{23} - C_{32}}{4q_4}, \quad q_2 = \frac{C_{31} - C_{13}}{4q_4}, \quad q_3 = \frac{C_{12} - C_{21}}{4q_4}.$$

If the trace is not positive, the computation focuses on the largest diagonal element to avoid numerical instability:

$$q_1 = \frac{1}{2}\sqrt{1 + C_{11} - C_{22} - C_{33}}, \quad \text{or} \quad q_2 = \frac{1}{2}\sqrt{1 - C_{11} + C_{22} - C_{33}}, \quad \text{or} \quad q_3 = \frac{1}{2}\sqrt{1 - C_{11} - C_{22} + C_{33}},$$

depending on which diagonal element is largest.

After calculation, the quaternion q is normalized to ensure unit length:

$$\|q\| = \sqrt{q_1^2 + q_2^2 + q_3^2 + q_4^2} = 1.$$

To avoid discontinuities in quaternion sign (quaternion double-cover property), if a previous quaternion q_{prev} exists, the dot product is used to flip the sign ensuring continuity:

$$\text{if} \quad q \cdot q_{\text{prev}} < 0, \quad q \leftarrow -q.$$

Direction Cosine Matrix (DCM) to Euler Angles

For the rotation sequence 1-2-3 (around z-y-x), Euler angles (ϕ, θ, ψ) correspond to roll, pitch, and yaw:

$$\begin{aligned} \theta &= \arcsin(C_{31}), \\ \phi &= \arctan 2(-C_{21}, C_{11}), \\ \psi &= \arctan 2(-C_{32}, C_{33}). \end{aligned}$$

Near the pitch singularities $\theta \approx \pm\frac{\pi}{2}$ (gimbal lock), standard formulas break down. The tool handles this by setting yaw $\psi = 0$ and computes roll ϕ differently to maintain a valid solution:

$$\phi = \arctan 2(C_{12}, -C_{13}), \quad \text{if} \quad \theta \approx +\frac{\pi}{2}$$

$$\phi = \arctan 2(C_{12}, C_{13}), \quad \text{if} \quad \theta \approx -\frac{\pi}{2}$$

Euler Angles to Direction Cosine Matrix (DCM)

Given Euler angles in degrees (ϕ, θ, ψ), they are first converted to radians. The DCM is constructed from the combined rotations (1-2-3 combination) about the fixed axes:

$$\mathrm{C} = \begin{bmatrix} \cos\theta\,\cos\psi & \cos\psi\,\sin\theta\,\sin\phi + \sin\psi\,\cos\phi & -\cos\psi\,\sin\theta\,\cos\phi + \sin\psi\,\sin\phi \\ -\sin\psi\,\cos\theta & -\sin\psi\,\sin\theta\,\sin\phi + \cos\psi\,\cos\phi & \sin\psi\,\sin\theta\,\cos\phi + \cos\psi\,\sin\phi \\ \sin\theta & -\cos\theta\,\sin\phi & \cos\theta\,\cos\phi \end{bmatrix}$$

Composite Conversions

This tool also supports direct conversions between Euler angles and quaternions by intermediate conversion via DCM:

$$\text{Euler Angles} \longrightarrow \text{DCM} \longrightarrow \text{Quaternion},$$

and

$$\text{Quaternion} \longrightarrow \text{DCM} \longrightarrow \text{Euler Angles}.$$

3 Free-torque propagation

Angular velocity vector propagation

In torque-free motion, no external torques act on the rigid body, resulting in a dynamic behavior governed solely by the body's initial angular velocity and inertia properties. Since the evolution of the angular velocity vector $\boldsymbol{\omega}$ depends only on itself and not on the orientation quaternion, we analyze $\boldsymbol{\omega}$ to validate the algorithm and study the natural rotational dynamics without external influence.

Figures Figure 1 and Figure 2 illustrate the time evolution of the components of the angular velocity and angular momentum vectors in the body frame, respectively. As predicted by theory, the absence of external torques leads to no net drift; each component oscillates periodically around a constant mean, maintaining the overall rotational kinetic energy and angular momentum magnitude. The dominant angular velocity component along the y-axis arises from the initial angular velocity distribution combined with the structure of the moment of inertia tensor $\mathbf{I}$.

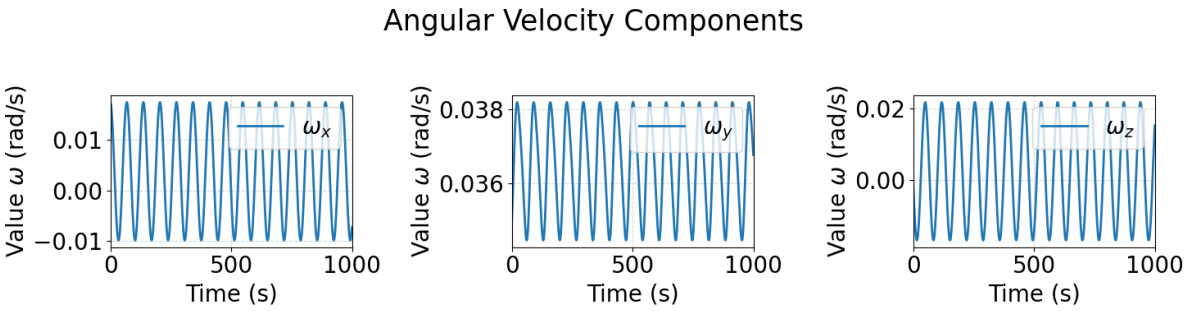

Figure 1: Time evolution of the angular velocity components (body frame)

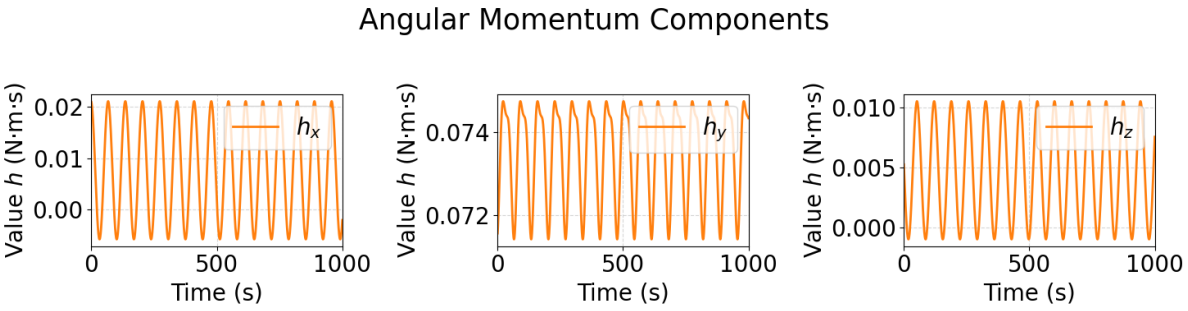

Figure 2: Time evolution of the angular momentum components (body frame)

In this free-torque regime, fundamental physical quantities remain conserved, expressed as:

Rotational kinetic energy:

$$T = \frac{1}{2}\boldsymbol{\omega}^{\mathrm{T}}\mathbf{I}\boldsymbol{\omega} = \text{constant}$$

Magnitude of the angular momentum:

$$\|\mathbf{h}\| = \|\mathbf{I}\boldsymbol{\omega}\| = \text{constant}$$

where $\mathbf{I}$ is the inertia tensor, $\boldsymbol{\omega}$ the angular velocity vector, and $\mathbf{h}$ the angular momentum vector. These conservation laws follow directly from Euler's equations in the absence of external

moments. To rigorously confirm the conservation, the percentage variation of these quantities over the simulation interval (0 to 1000 seconds) is computed as:

$$\text{percent\_variation}_{T(t)} = \frac{\max(T(t)) - \min(T(t))}{\max(T(t))} \times 100 = 1.72 \times 10^{-8}\%$$

$$\text{percent\_variation}_{\|\mathbf{h}(t)\|} = \frac{\max(\|\mathbf{h}(t)\|) - \min(\|\mathbf{h}(t)\|)}{\max(\|\mathbf{h}(t)\|)} \times 100 = 6.74 \times 10^{-9}\%$$

These near-zero variations demonstrate that both the rotational kinetic energy and angular momentum magnitude remain effectively constant throughout the free-torque motion, in excellent agreement with theoretical expectations. This validates the numerical algorithms and confirms the physical fidelity of the simulation.

A final proof comes from the representation of the kinetic energy and angular momentum ellipsoids, expressed as:

Kinetic energy ellipsoid:

$$\frac{\omega_x^2}{2T_i/I_x} + \frac{\omega_y^2}{2T_i/I_y} + \frac{\omega_z^2}{2T_i/I_z} = 1$$

Angular momentum sphere:

$$h_x^2 + h_y^2 + h_z^2 = \|\mathbf{h}_i\|^2$$

Here, T_i and $\mathbf{h}_i$ are the initial kinetic energy and angular momentum vector in the body frame, and I_x, I_y, I_z are the principal moments of inertia.

These equations are valid strictly in the principal axis frame; therefore, to perform the representations, the vectors must first be transformed to the principal axes and then transformed back to the body frame. Finally, both ellipsoids are represented in the angular momentum space.

From theory, an intersection between these surfaces is expected. This intersection represents the polhode curve of the angular velocity vector in the body frame, reflecting the simultaneous conservation of kinetic energy and the magnitude of angular momentum during free-torque motion.

As shown in Fig. 3, the propagated angular momentum vector (red line) traces the intersection between the kinetic energy ellipsoid (green surface) and the angular momentum sphere (blue surface). This confirms the effectiveness of the algorithm and its consistency with theoretical expectations.

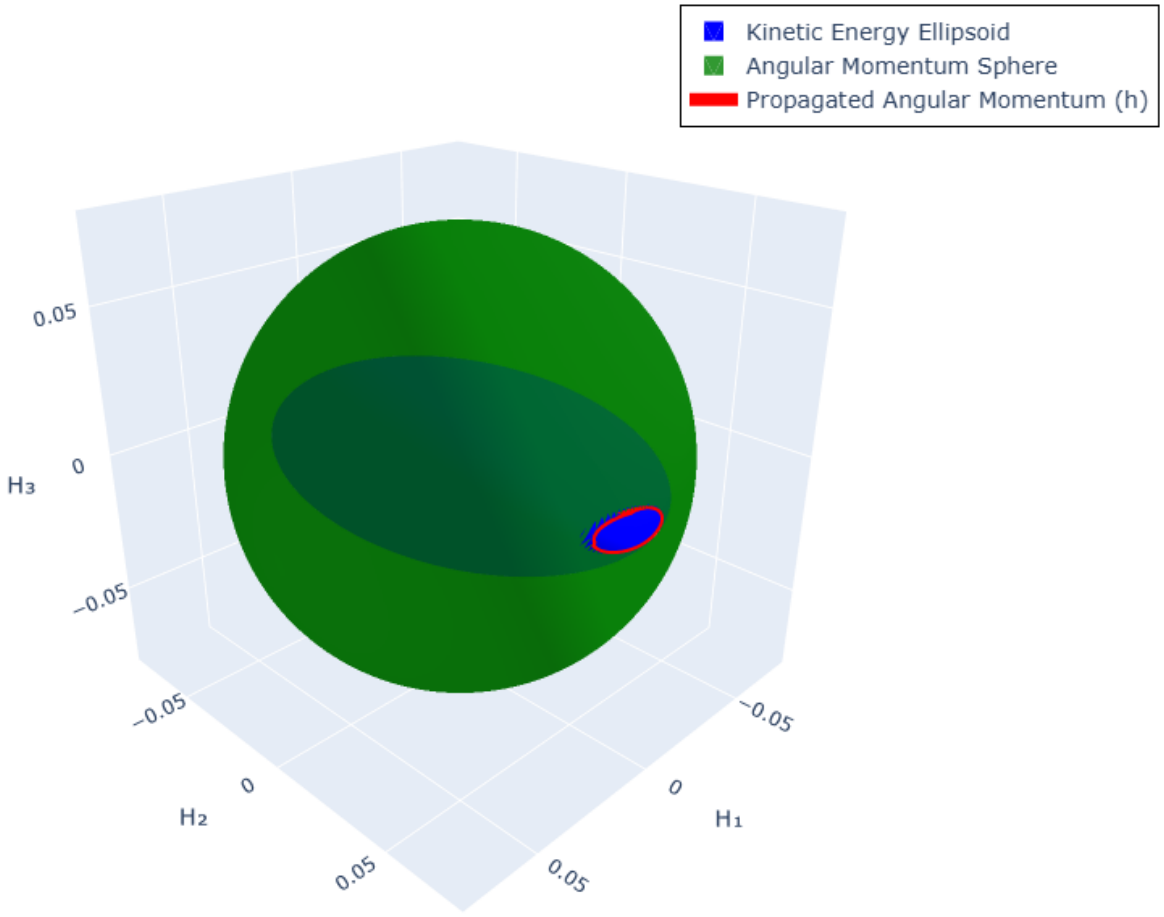

Figure 3: Kinetic energy ellipsoid and angular momentum sphere

Quaternion propagation

The initial quaternion, expressed in the sensor frame, is rotated to align with the body frame. The corresponding direction cosine matrix (DCM) for the required transformation from the sensor to the body frame is given by:

$$\mathbf{C}_{\text{sensor}\to\text{body}} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

The rotation is applied using quaternion multiplication, as described in section 1, resulting in:

$$\mathbf{q}_{\text{inertia}\to\text{body}} = \begin{bmatrix} -0.02085965 & -0.75043611 & 0.32660553 & 0.57422935 \end{bmatrix}$$

The propagated quaternion is shown in Figure 4. Since quaternions themselves lack direct physical interpretability, the result is converted into unwrapped Euler angles to provide a clearer understanding of the satellite's angular evolution (see Figure 5). The Euler angle plots reveal that the rotations about the x and y axes oscillate without drift, while the rotation about the z axis exhibits a steady accumulation, leading to large angular values over time.

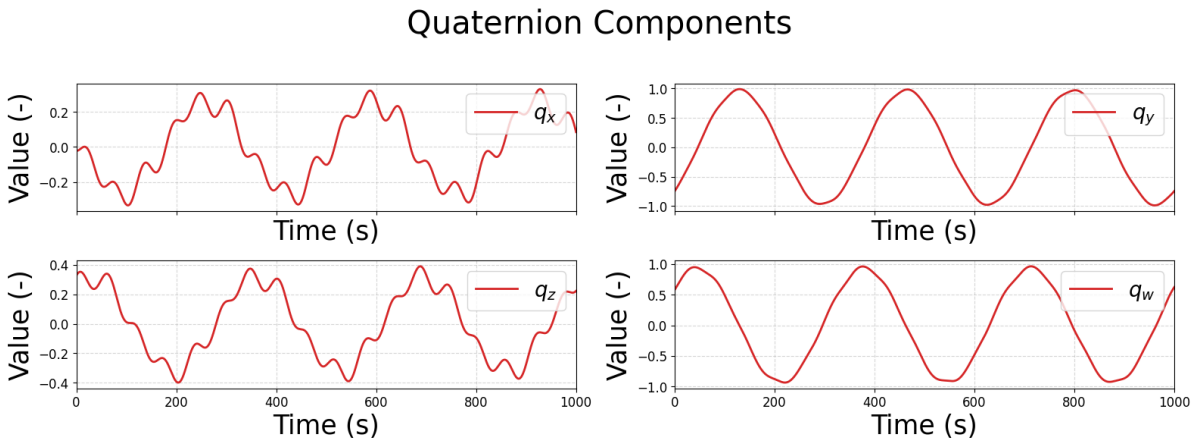

Figure 4: Time evolution of the quaternion (inertial to body frame)

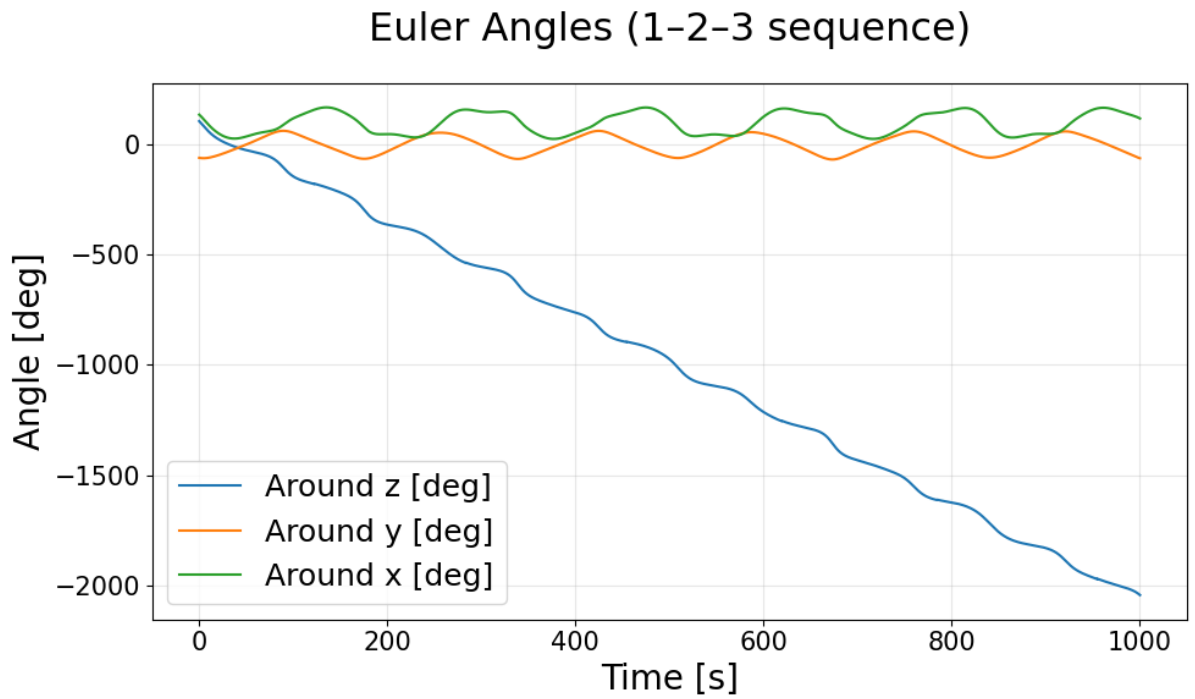

Figure 5: Time evolution of the Euler angles (inertial to body frame)

4 Constant Torque Propagation

Angular velocity vector propagation

As in section 3, the propagation of the angular velocity components is first analyzed to assess the accuracy of the algorithm under constant torque. The time evolution of the angular velocity and angular momentum components are illustrated in Figure 6 and Figure 7. The constant torque produces a drift in the output, so that the components oscillate while drifting rather than oscillating around a fixed mean. The most prominent drift appears in the y-component, influenced by the initial conditions, inertia matrix and external torque.

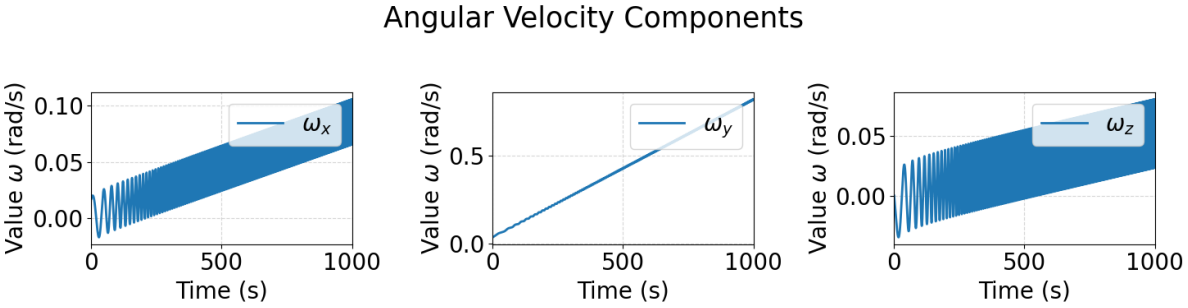

Figure 6: Time evolution of the angular velocity components (body frame)

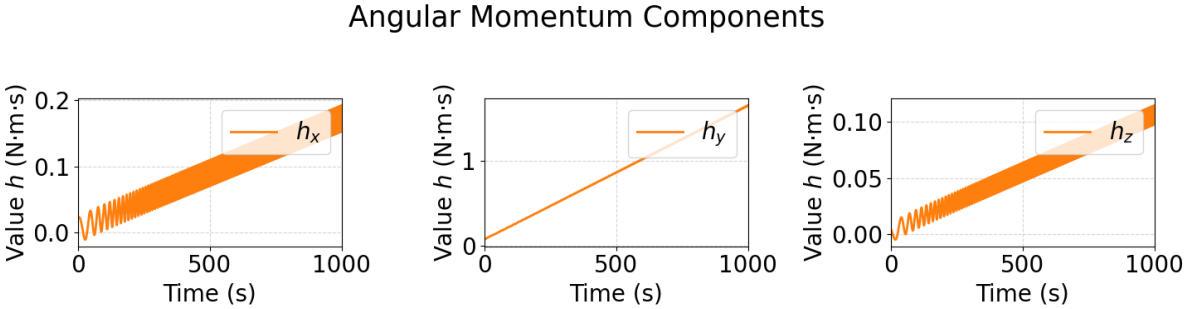

Figure 7: Time evolution of the angular momentum components (body frame)

This analysis shows that the main drift occurs in the y component, as expected from the torque orientation and inertia distribution. The algorithm thus correctly captures the coupled dynamics, with each component's evolution aligning with theoretical predictions.

In the presence of an external torque, neither the norm of the angular momentum vector nor the kinetic energy remains conserved. This is reflected in the observed variations during the simulation:

Percent variation in $T(t)$:

$$\text{percent\_variation}_{T(t)} = \frac{\max(T(t)) - \min(T(t))}{\max(T(t))} \times 100 = 99.8\,\%$$

Percent variation in $\|\mathbf{h}(t)\|$:

$$\text{percent\_variation}_{\|\mathbf{h}(t)\|} = \frac{\max(\|\mathbf{h}(t)\|) - \min(\|\mathbf{h}(t)\|)}{\max(\|\mathbf{h}(t)\|)} \times 100 = 95.5\,\%$$

As illustrated in Figure 8, the angular momentum trajectory (red line) is shown together with the kinetic energy ellipsoid (green surface) and the angular momentum sphere (blue surface). While the motion begins on the polhode, that is, on the intersection of these two surfaces, the continuous action of external torque drives the angular momentum away from the intersection curve. This visually confirms the loss of conservation for both kinetic energy and angular momentum magnitude in the presence of external torques.

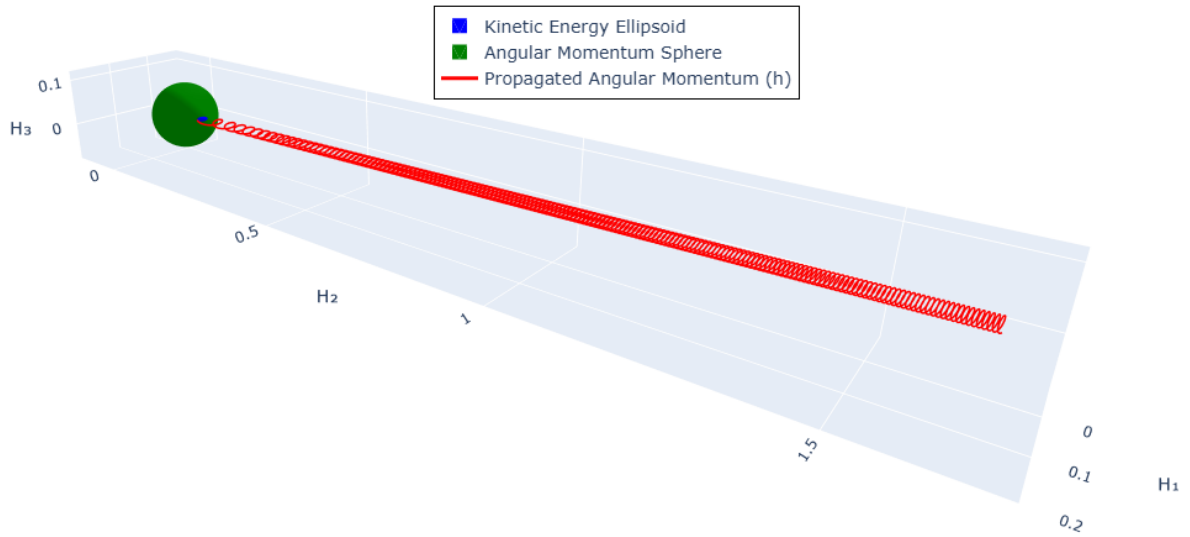

Figure 8: Kinetic energy ellipsoid and angular momentum sphere, illustrating the angular momentum trajectory under constant external torque.

Quaternion propagation

The resulting time histories of the quaternion and Euler angles are shown in Figure 9 and Figure 10. Under a constant torque, the magnitude of $\boldsymbol{\omega}$ generally increases with time, so the instantaneous rotation rate grows. Consistent with this, the quaternion components display oscillations whose period shortens over time; equivalently, their oscillation frequency increases as time progresses.

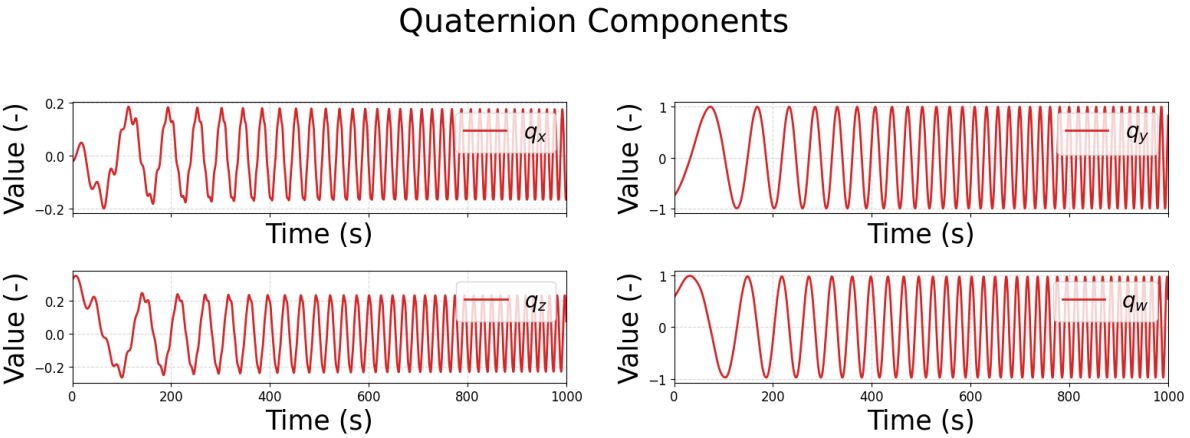

Figure 9: Time evolution of the quaternion (inertial to body frame)

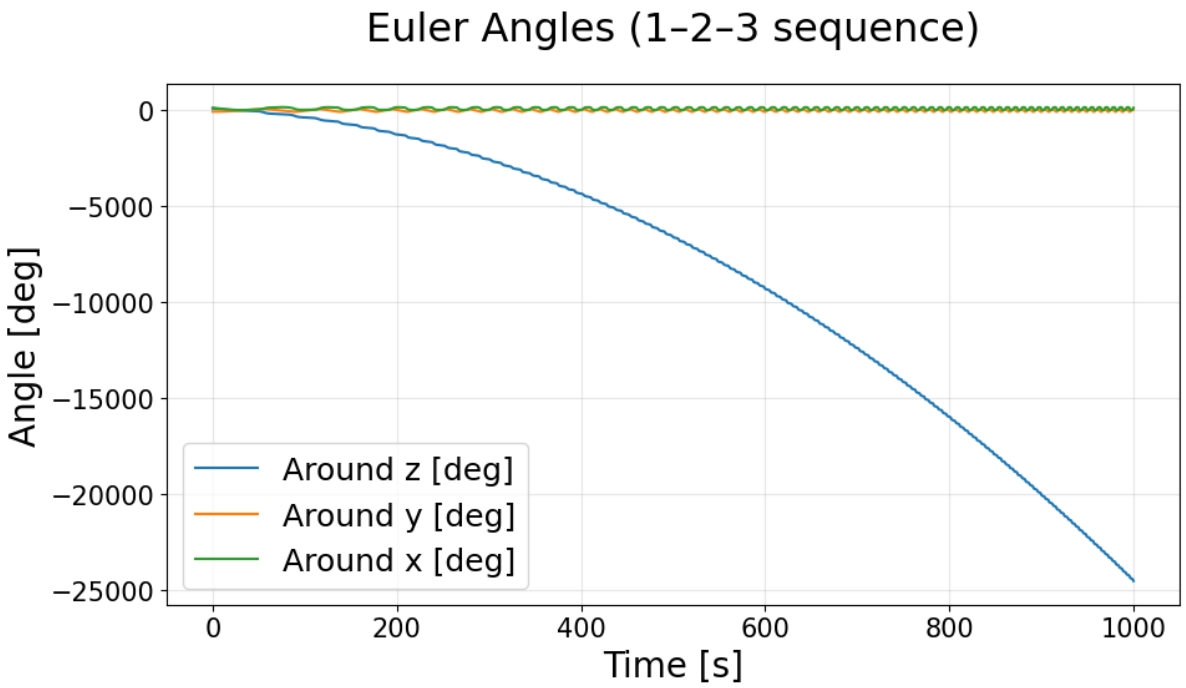

Figure 10: Time evolution of the Euler angles (inertial to body frame)

For the Euler-angle representation, the x- and y-axis angles exhibit an increasing oscillation frequency over time, reflecting the growth of $\|\boldsymbol{\omega}\|$. In contrast, the z-axis angle shows an accelerated drift with respect to the torque-free case.