Interactive Rotational Dynamics

From Euler Angles to Quaternions: A Visual Journey

14.01.2026

Pablo Hernández

Programando todo tipo de cosas desde 2010

Mi stack

Actualmente estudiando en ULL

hadronomy.com hadronomy hadronomy

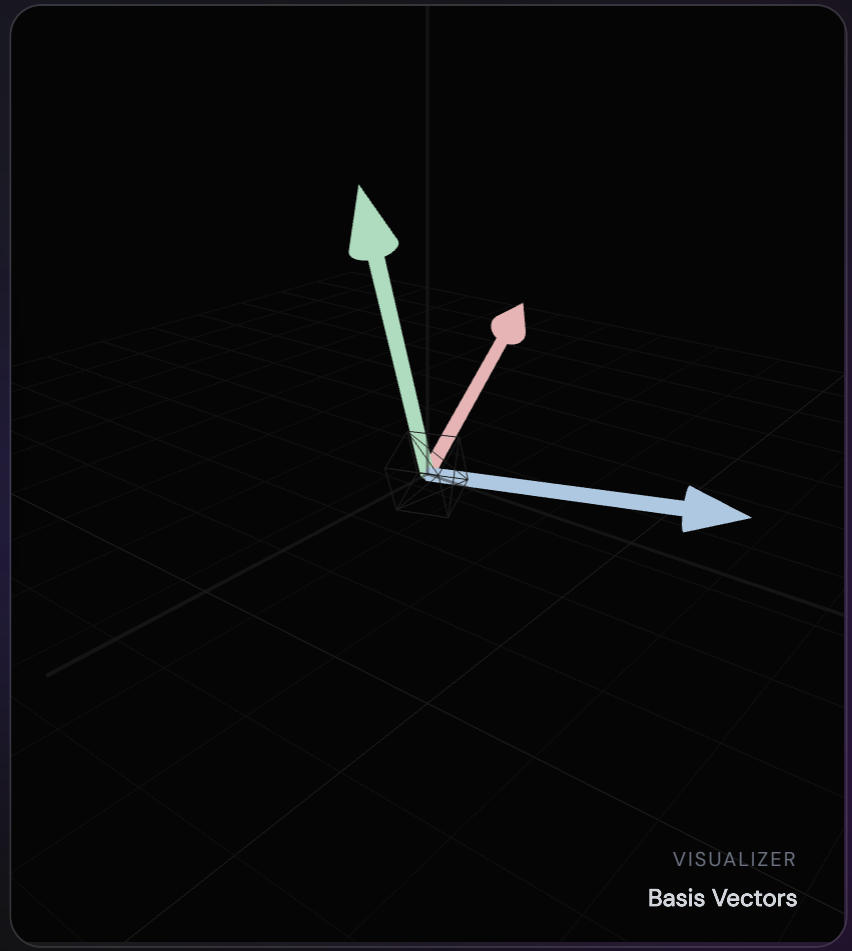
SO(3) Groups

Special Orthogonal Group in 3D

BASIS X	BASIS Y	BASIS Z
-0.26	0.03	0.97
0.32	0.95	0.05
-0.91	0.32	-0.25

The "Aha!" Moment: A rotation matrix isn't just random numbers.
Column 1 is simply the coordinates of the Red Arrow.
Column 2 is the Green Arrow.

PAUSE ROTATION



Euler Angles & Gimbal Lock

The Jacobian Singularity

OUTER GIMBAL (PITCH) 0°

MIDDLE GIMBAL (YAW) 0°

Safe Safe

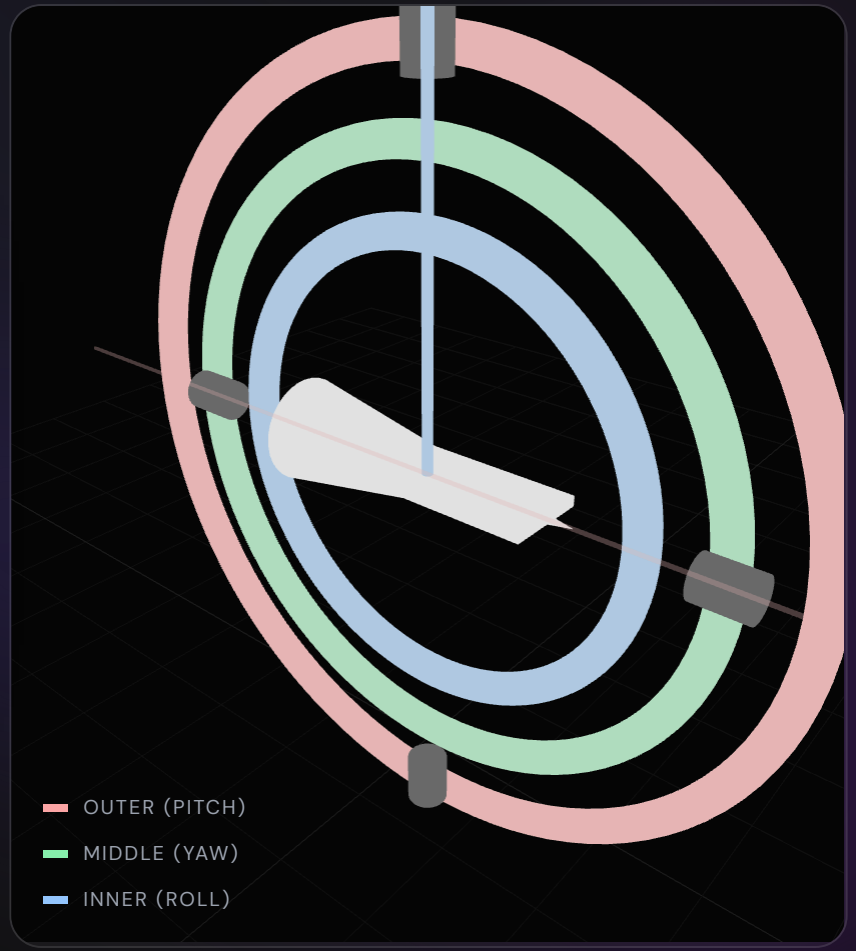
INNER GIMBAL (ROLL) 0°

DETERMINANT

1.000

AXIS ALIGNMENT

SNAP TO LOCK



Anatomy of a Quaternion

$$q = w + xi + yj + zk$$

EULER'S ROTATION FORMULA

w = cos(θ/2) **1.000**

xyz = û · sin(θ/2)

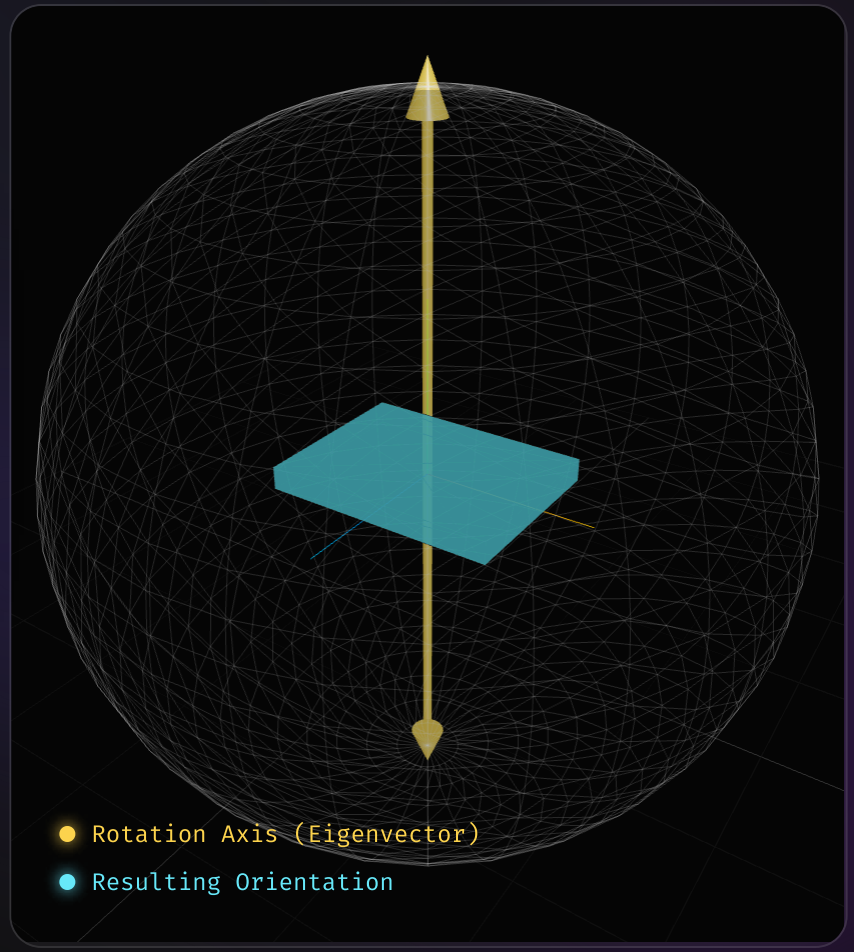
x 0.000
y 0.000
z 0.000

ROTATION ANGLE (ϴ) 0°

0° 360° 720°

ROTATION AXIS (Û)

Vertical (Latitude) Horizontal (Longitude)



The Double Cover of SO(3)

PHYSICAL ROTATION

0°

TOPOLOGICAL STATE

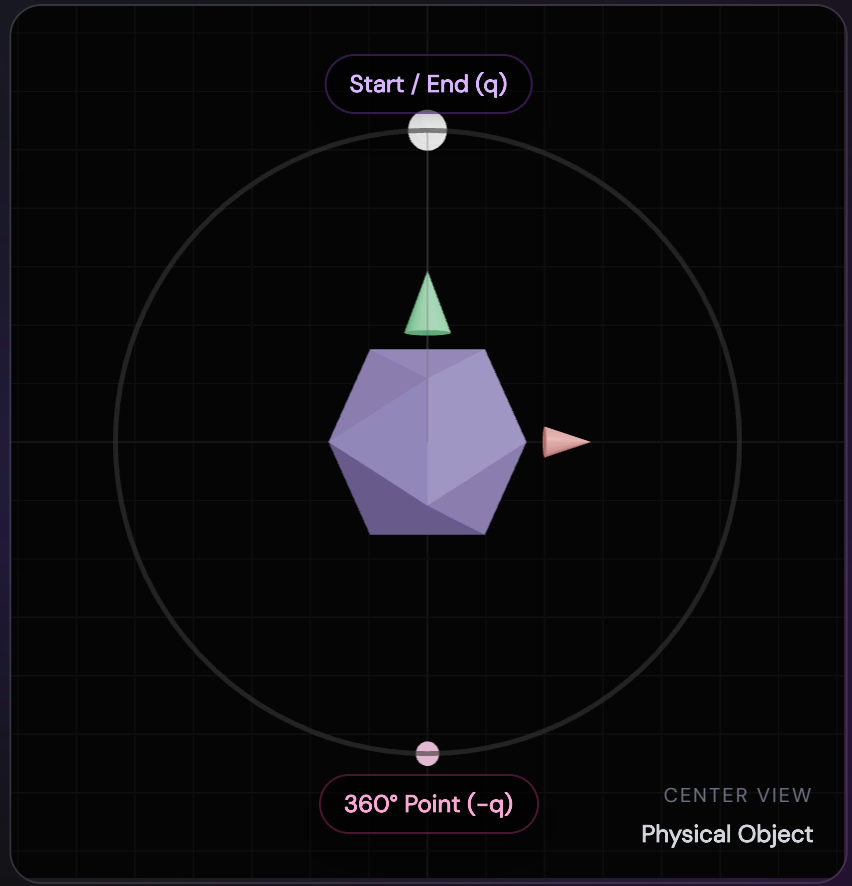
Identity (q)

Watch the **Configuration Ring** on the right.

- At **360°**, the object looks reset, but the math state is antipodal **(-q)**.
- It takes **720°** (two spins) to fully return to the identity state **(q)**.

0° (Q) 360° (-Q) 720° (Q)

▶ VISUALIZE 720° CYCLE



The Sandwich Product

$$p' = q \cdot p \cdot q^{-1}$$

$$q \cdot p \cdot q^{-1}$$

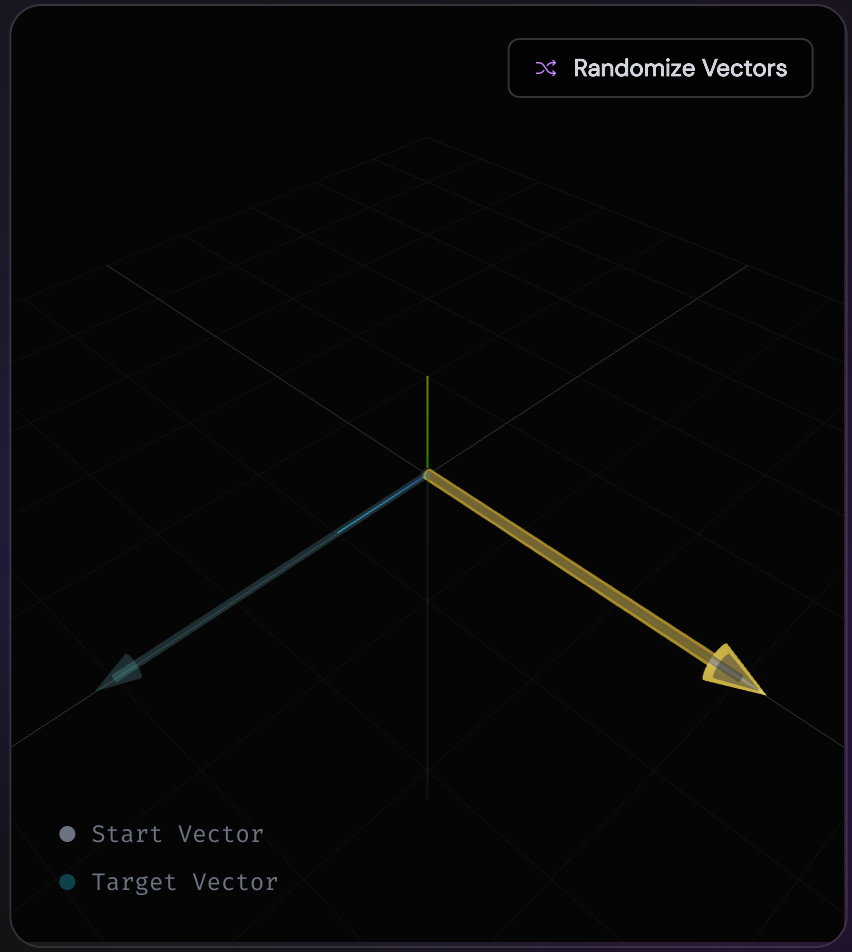
W: 0.71 XYZ: [0.00, -0.71, 0.00] ANGLE: 90°

0 — 1 — 2

Initial State

Vector p ready for rotation.

NEXT STEP →

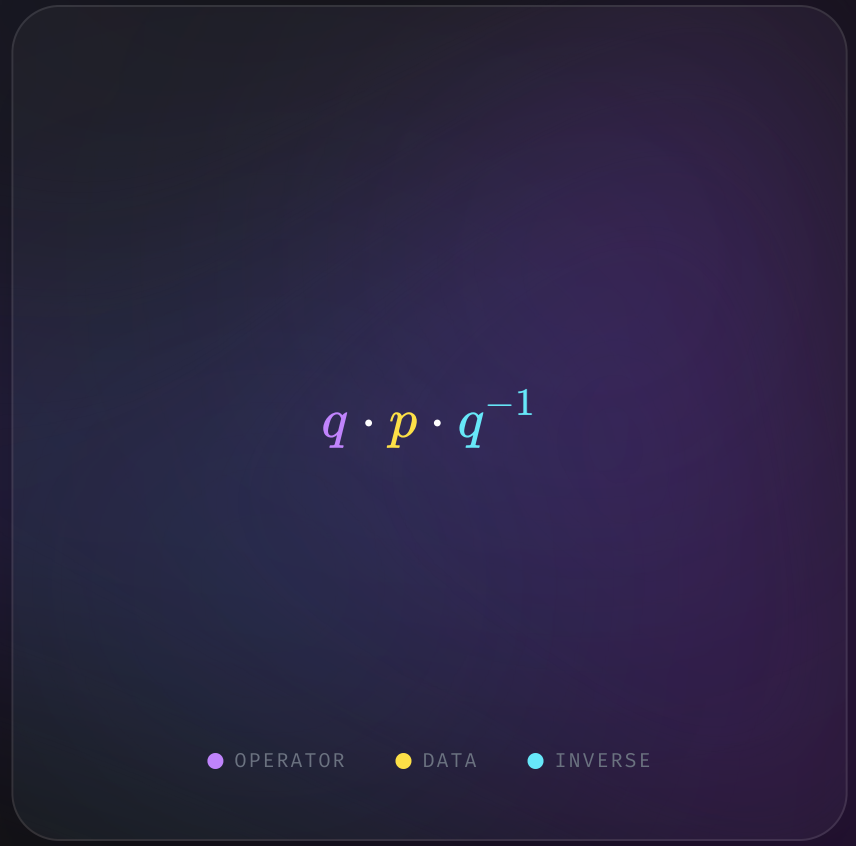


Algebraic Mechanics

Why does it work?

THE EQUATION

The standard formula for rotating a vector using quaternions.



STEP 1 / 4

Algebraic Mechanics

Why does it work?

1. ENCODING (PURE QUATERNION)

We promote the 3D vector **v** into 4D space by setting the scalar part (w) to 0.

$$p = [0, \mathbf{v}]$$

OPERATOR DATA INVERSE

Algebraic Mechanics

Why does it work?

2. THE FIRST PRODUCT (TWIST)

Multiplying q and p creates a 'Spinor'. Notice the scalar part is **non-zero**. It is no longer a vector; it is a mixed 4D state.

$$qp = [-\mathbf{u} \cdot \mathbf{v}, \\ w\mathbf{v} + \mathbf{u} \times \mathbf{v}]$$

OPERATOR DATA INVERSE

Algebraic Mechanics

Why does it work?

3. THE INVERSE PRODUCT (UNTWIST)

The second multiplication cancels out the scalar part exactly, returning a 'Pure Quaternion'. The vector has been rotated by 2θ.

$$(qp)q^{-1} = [0, \mathbf{v}']$$

OPERATOR DATA INVERSE

SLERP vs NLERP

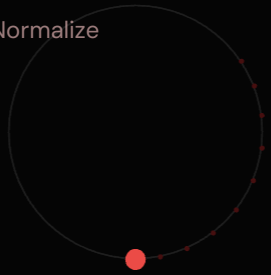
Visualizing Angular Velocity Variance

SPEED

RACE

NLERP

Linear Interpolation + Normalize

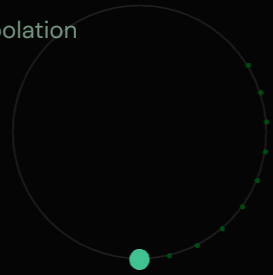


VELOCITY

28 deg/s

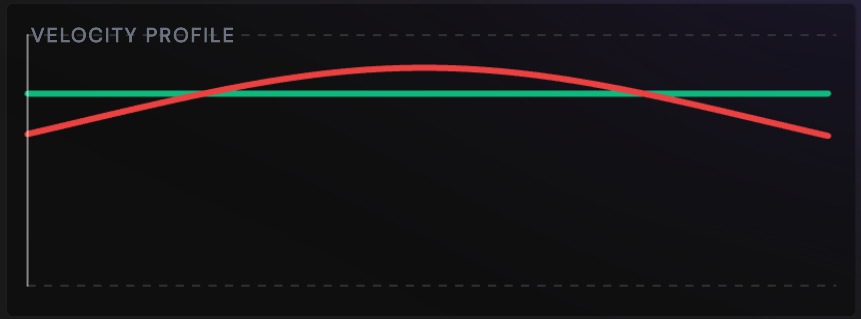
SLERP

Spherical Linear Interpolation



VELOCITY

32 deg/s



NLERP Variable

Shortcuts through the sphere's chord. Velocity peaks at 50% progress (midpoint acceleration), creating unnatural motion.

SLERP Constant

Calculates the exact great-circle arc. Maintains perfect angular velocity throughout the rotation.

Computational Decay

Visualizing the loss of geometric integrity

SIMULATE DECAY

Rotation Matrix

9 Floats (Unconstrained)

VOLUME (DETERMINANT)
Target: 1.00

1.000

Quaternion

4 Floats (Normalized)

VOLUME (DETERMINANT)
Preserved via normalization

1.000

THE PROBLEM

Rotation matrices accumulate floating point errors in 9 separate values. The axes stop being 90° apart (Shear) and change length (Scale).

Result: The geometry is crushed.

THE FIX

Quaternions only have 4 values. Drift creates scale error, but `normalize(q)` costs only 4 multiplications.

Result: Perfect rotation forever.

1.5x

PERFORMANCE GAIN

In physics engines

Thank You

Slides and references at: hadronomy.com,
video at youtube