THE UNIVERSITY *of* EDINBURGH
informatics

Advanced Robotics

SO(3)

Steve Tonneau
School of Informatics
University of Edinburgh

Summary and todos

An element of the group….

Can be represented as …

$r \in SO(3)$ $\simeq$ $\mathbf{R} \in \mathbb{R}^{3\times 3}$ Rotation matrix

rotation

$\boxed{\begin{array}{ll} \mathbf{q} \in \mathbb{H} \simeq \mathbb{R}^4, \|\mathbf{q}\| = \mathbf{1} & \text{quaternion} \\ \mathbf{w} \in so(3) \simeq \mathbb{R}^3 & \text{A velocity ?} \end{array}}$

$m \in SE(3)$ $\simeq$ $\mathbb{R}^3 \times SO(3)$ $\simeq$ $\mathbb{R}^3 \times \mathbb{R}^{3\times 3} \simeq \mathbb{R}^{4\times 4}$ Homogeneous matrix

displacement

translation rotation

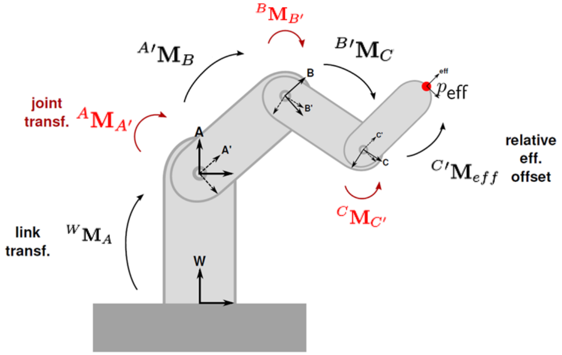
$\boxed{\begin{array}{ll} \mathbb{R}^3 \times \mathbb{H} \simeq \mathbb{R}^7 & \\ \mathcal{V} \in se(3) = \begin{bmatrix} \mathbf{v} \\ \mathbf{w} \end{bmatrix} \simeq \mathbb{R}^6 & \text{A spatial velocity ?} \end{array}}$

Why is it so complicated to represent a rotation ?
Answer tomorrow

Remember what we need

- Composition

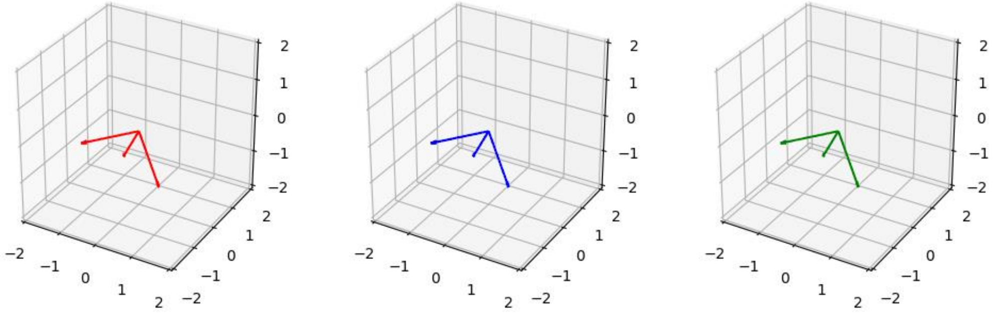
  Matrices work well with that



- But also interpolation

  Unfortunately interpolation of rotation matrices do not result in a rotation matrix.

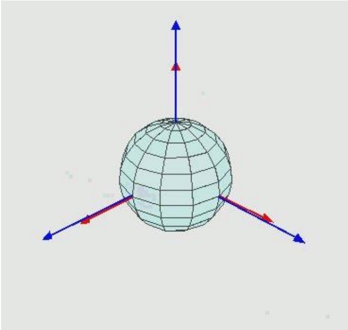
$$R^{-1} = R^{\top}$$
$$\det(R) = 1$$



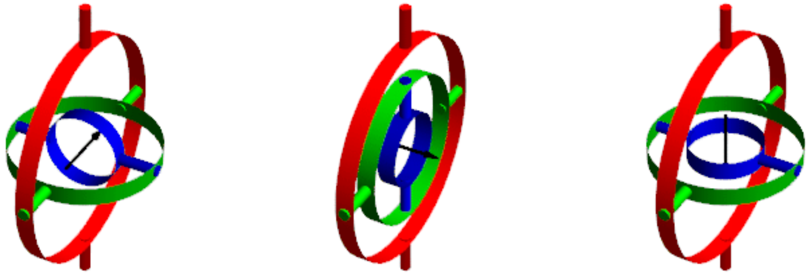
Can we represent SO(3) with only 3 parameters?

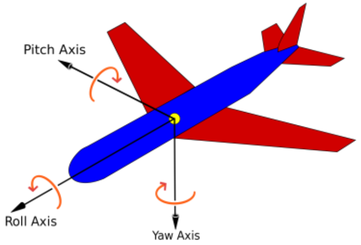
- ❑ Euler angles: 3 consecutive unit rotations around different axes.
  - ❑ Many conventions possible:

    Z-Y-Z (3-1-3) representation



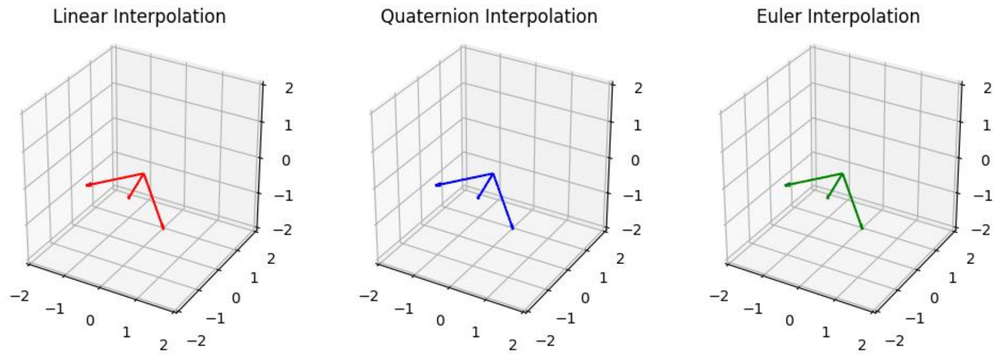
    yaw-pitch-roll or Z-Y-X (3-2-1) --> used in flight!





Euler Angles and Gimbal Lock

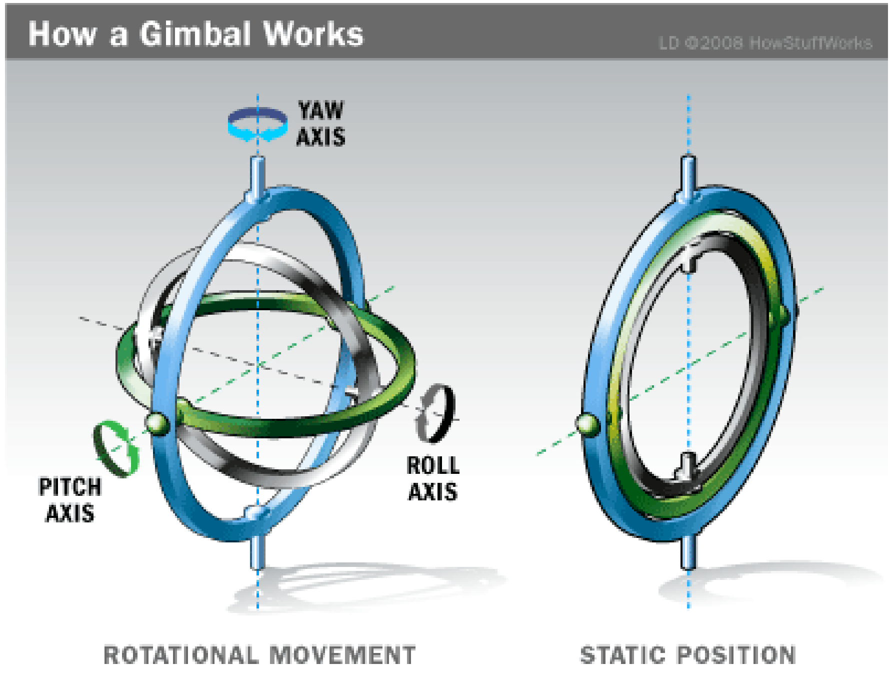
- Euler angles have a severe problem
  - If two axes align: blocks 1 DOF
  - ‘singularity’ of Euler angles



- Pros:
  - minimal representation
  - human readable

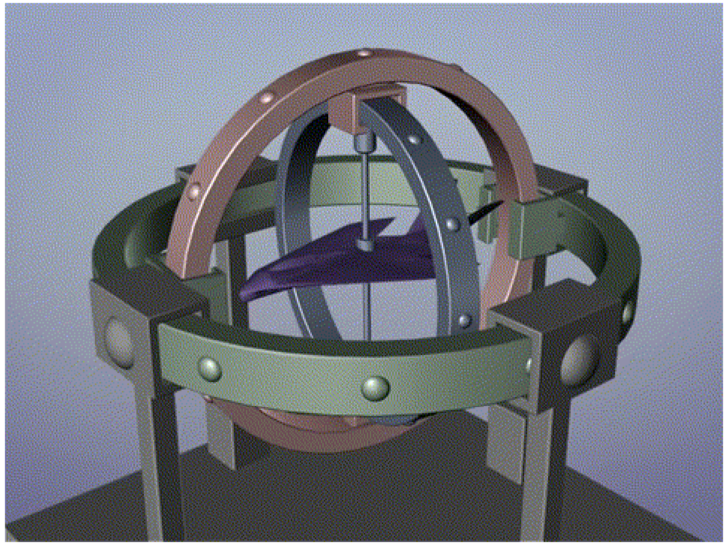
- Cons:
  - Gimbal lock
  - Infinite ways to represent rotation (mod. 2Π)
  - must convert to matrix to rotate vector
  - no easy composition
  - Interpolation does not give shortest path

Euler Angles and Gimbal Lock



https://compsci290-s2016.github.io/CoursePage/Materials/EulerAnglesViz/

- Cons: the ***Gimbal Lock***



See source for a good explanation on consequences / mitigation https://www.youtube.com/watch?v=zc8b2Jo7mno

Gimbal lock from matrix operation point of view

(from Wikipedia: https://en.m.wikipedia.org/wiki/Gimbal_lock#Loss_of_a_degree_of_freedom_with_Euler_angles)

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

An example worth examining happens when $\beta = \frac{\pi}{2}$. Knowing that $\cos\frac{\pi}{2} = 0$ and $\sin\frac{\pi}{2} = 1$, the above expression becomes equal to:

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Carrying out matrix multiplication:

$$R = \begin{bmatrix} 0 & 0 & 1 \\ \sin\alpha & \cos\alpha & 0 \\ -\cos\alpha & \sin\alpha & 0 \end{bmatrix} \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ \sin\alpha\cos\gamma + \cos\alpha\sin\gamma & -\sin\alpha\sin\gamma + \cos\alpha\cos\gamma & 0 \\ -\cos\alpha\cos\gamma + \sin\alpha\sin\gamma & \cos\alpha\sin\gamma + \sin\alpha\cos\gamma & 0 \end{bmatrix}$$

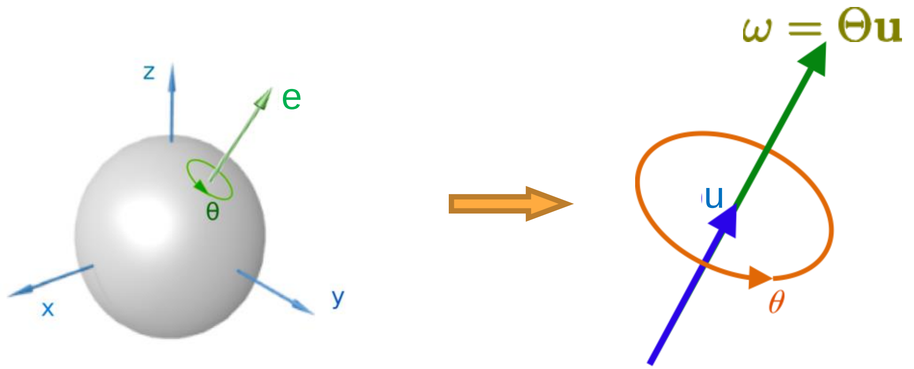
And finally using the trigonometry formulas:

$$R = \begin{bmatrix} 0 & 0 & 1 \\ \sin(\alpha+\gamma) & \cos(\alpha+\gamma) & 0 \\ -\cos(\alpha+\gamma) & \sin(\alpha+\gamma) & 0 \end{bmatrix}$$

Changes in α and γ now affect the same rotation axis

Angle axis representation

- In 3D space, any rotation of a rigid body is equivalent to a single rotation by a given angle θ about a fixed axis u (ie, the Euler axis).



- Issue: again, not a map, ie hard to rotate a point directly

- However, this will be important later. Let’s do some maths to show how this relate to angular velocities

Angle axis derivation

Angle axis derivation

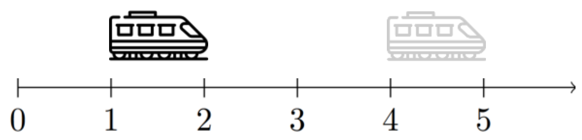
How does velocity represent a rotation?

$$r \in SO(3) \qquad \simeq \qquad \mathbf{w} \in so(3) \simeq \mathbb{R}^3$$

- ❑ $so(3)$ is a Lie algebra (just name-dropping here, more later)

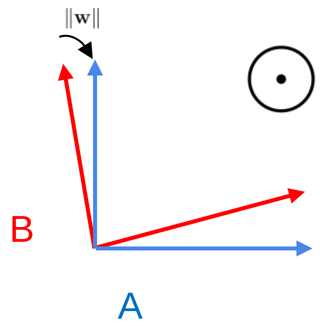
- ❑ r obtained by integrating $\mathbf{w}$ for a duration of 1 (no unit)

Easy to see in linear space, though not so interesting



$\mathbf{w} = [3]$

To align A and B we integrate a constant velocity during a time of 1



$\mathbf{w} = [0, 0, \theta]^T$

Angle axis derivation

- ❑ $\mathbf{w} \in \mathbb{R}^3$ is a minimal representation of $\boldsymbol{r} \in \boldsymbol{SO(3)}$

- ❑ However it does not act as a map. Using velocity definition and derivative we can retrieve matrix representation through matrix exponentiation (Rodrigues formula) - see notes in drupal for full derivation:

$$\mathbf{R}(\mathbf{w}) = \mathbf{I} + sin\theta\mathbf{u}_\times + (1 - cos\theta)\mathbf{u}_\times^2$$

where

$$\mathbf{w} = \theta\mathbf{u}, \mathbf{u} = [u_x, u_y, u_z]^T, ||\mathbf{u}||_2 = 1$$

$$\mathbf{u}_\times = \begin{bmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{bmatrix}$$

Skew-symmetric matrix, encoding cross product by **u**

Angle axis summary

- **w** minimal representation or rotation
- Can't be used as operator, needs conversion to matrix form

$$\mathbf{R} = \exp(\mathbf{w}_\times)$$

- A log operator also exists

$$\mathbf{w}_\times = \log(\mathbf{R})$$

- Can we find a 3D operator?

A minimal 3D tool for rotations?

A minimal 3D tool for rotations?

3D extension of complex numbers ?

- Something like $r = a + ib + jc$

- Such that composition would work:

  $r_1 * r_2 \Leftrightarrow$ apply transformation r_1, then r_2 [1]

- What rules for j such that composition works ?

It has been proven that no such representation exists in 3D!

[1] More formally and generally, looking for an unital asociative algebra A such that $(r_1 * r_2) * r_3 = r_1 * (r_2 * r_3)$ $\forall r_k$ (associativity) and $1 * r = r = r * 1$ $\forall r$ (unital)

A minimal 3D tool for rotations?

3D extension of complex numbers ?

- Something like $r = a + ib + jc$

- Such that composition would work:

  $r_1 * r_2 \Leftrightarrow$ apply transformation r_1, then r_2 [1]

- What rules for j such that composition works ?

[1] More formally and generally, looking for an unital asociative algebra A such that $(r_1 * r_2) * r_3 = r_1 * (r_2 * r_3)$ $\forall r_k$ (associativity) and $1 * r = r = r * 1$ $\forall r$ (unital)

*Quaternions

Quaternions: extension of complex numbers from 2 dimensions, into 4 dimensions.

Complex numbers: $i^2 = -1, \text{ie} \quad a + b \cdot i$

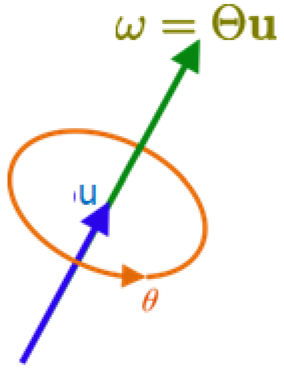
Quaternions: $i^2 = j^2 = k^2 = i \cdot j \cdot k = -1 \cdot$

$\text{ie} \quad a + b \cdot i + c \cdot j + d \cdot k$

×	**1**	***i***	***j***	***k***
1	1	i	j	k
i	i	-1	k	$-j$
j	j	$-k$	-1	i
k	k	j	$-i$	-1

Those are the conditions for the Algebra to be unital and associative

Recall the Axis-Angle representation:



1) 3d rotation axis is defined by a vector (position in [m]);

2) rotational angle is defined by radians.

Quaternions represent all 4 dimensions coherently.

*Rotation: Quarternion

A quaternion is $r \in \mathbb{R}^4$

$$\mathbf{r} = \begin{bmatrix} sin(\theta/2)u_1 \\ sin(\theta/2)u_2 \\ sin(\theta/2)u_3 \\ cos(\theta/2) \end{bmatrix}$$

with **u** = [u_1, u_2, u_3] unit rotation axis

Unit length constraint (to represent rotations)

$$r^\top r = r_0^2 + r_1^2 + r_2^2 + r_3^2 = 1$$

A single rotation can be defined by two different quaternions (this will matter!)

Exercise:
What is the equivalent quaternion?

A map can be obtained to apply a rotation r to p

$$\mathbf{r_{p'}} = \mathbf{r}\mathbf{r_p}\bar{\mathbf{r}}$$

With r_p= [p_x, p_y, p_z, 0] (pure quaternion)

*Quaternions: Pros and Cons

- Pros: mini rotation matrices
  - no singularity
  - almost minimal representation
  - easy to enforce constraints
  - easy composition
  - easy interpolation
- Cons
  - somewhat confusing

- **Summary** of Rotation representations
  - need rotation matrix for the rotational operation
  - Quarternions good for free rotations
  - Euler angles OK for small angular deviations…
    - but beware of singularities! I don’t recommend it