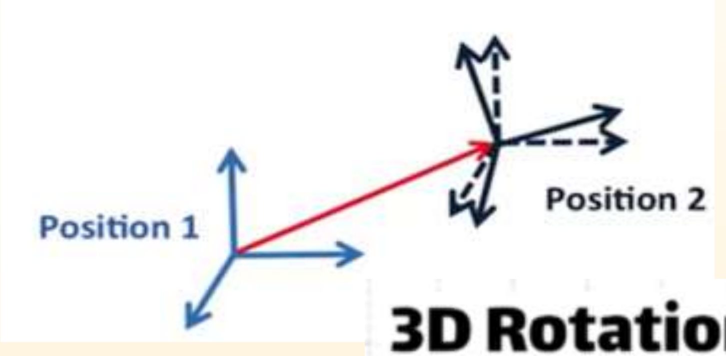
Fundamental of 3D Motion

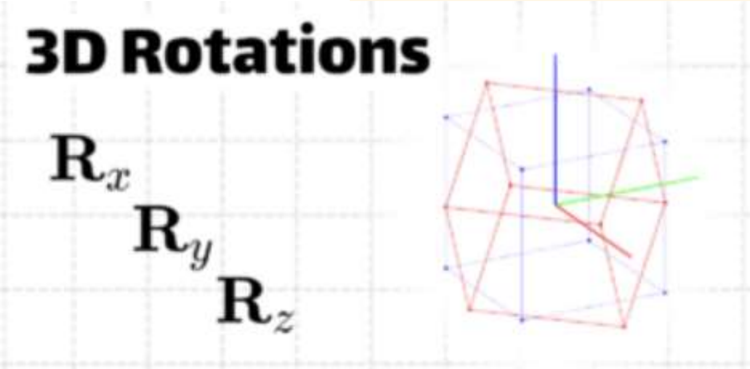
By
Weitao Xiong

Advisor:
Prof. Hongfei Xue

3D Motion

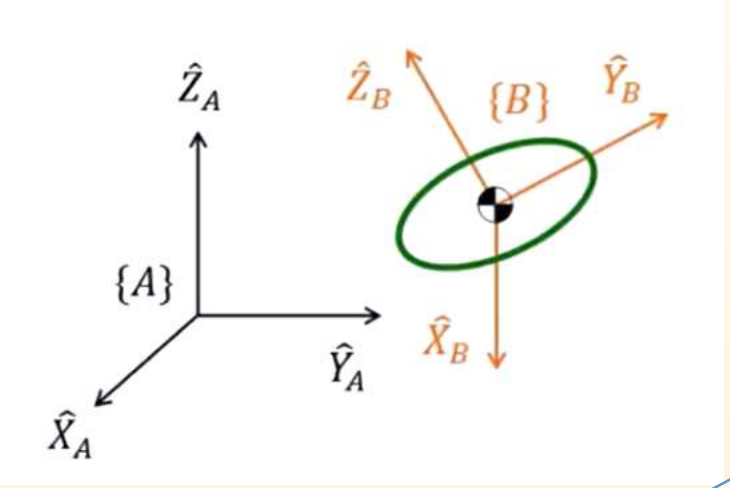
- 3D Motion consists of
  - ✓ Translation
  - ✓ pure rotation
- Rotation is complex, it focus on
  - ✓ Rotation Matrix
  - ✓ Euler Angles
  - ✓ Gimbal Lock
  - ✓ Quaternion
  - ✓ Axis-Angle





Rotation Matrix

- First, it can describe the posture of {B} relative to {A}, {A} is global axis



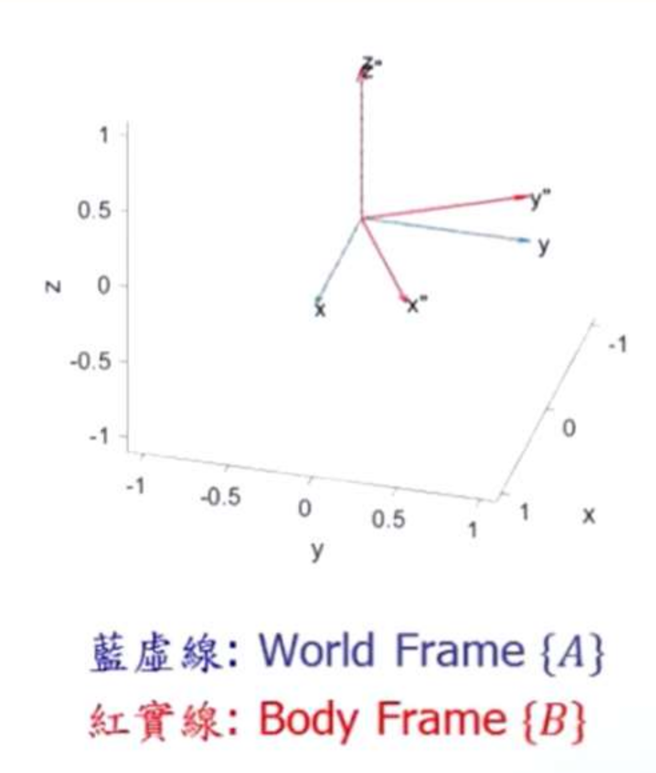
- Suppose each length of vectors is 1

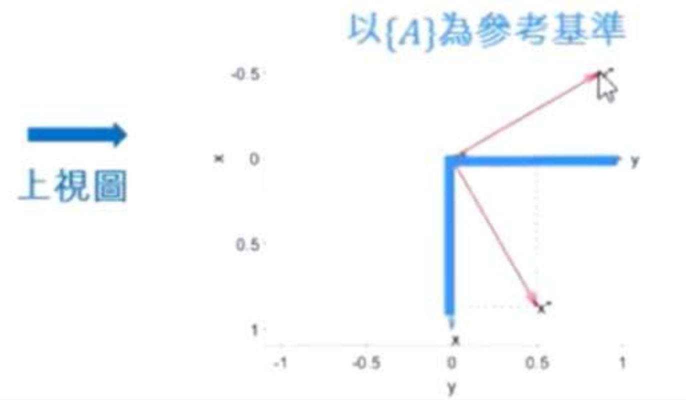
Means the Projection of X_B on {A}

$$
{}^{A}\mathbf{R}_{B} = \begin{bmatrix} \boxed{\begin{matrix} \hat{X}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A \end{matrix}} & \begin{matrix} \hat{Y}_B \cdot \hat{X}_A \\ \hat{Y}_B \cdot \hat{Y}_A \\ \hat{Y}_B \cdot \hat{Z}_A \end{matrix} & \begin{matrix} \hat{Z}_B \cdot \hat{X}_A \\ \hat{Z}_B \cdot \hat{Y}_A \\ \hat{Z}_B \cdot \hat{Z}_A \end{matrix} \end{bmatrix} = \begin{bmatrix} | & | & | \\ {}^{A}\hat{X}_B & {}^{A}\hat{Y}_B & {}^{A}\hat{Z}_B \\ | & | & | \end{bmatrix}
$$

"direct cosines"

Ex:





$$
{}^A\hat{X}_B = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A \end{bmatrix} = \begin{bmatrix} 1 \times \frac{\sqrt{3}}{2} \\ 1 \times \frac{1}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} 0.866 \\ 0.5 \\ 0 \end{bmatrix}
$$

$$
{}^A\hat{Y}_B = \begin{bmatrix} \hat{Y}_B \cdot \hat{X}_A \\ \hat{Y}_B \cdot \hat{Y}_A \\ \hat{Y}_B \cdot \hat{Z}_A \end{bmatrix} = \begin{bmatrix} -0.5 \\ 0.866 \\ 0 \end{bmatrix}
$$

$$
{}^A\hat{Z}_B = \begin{bmatrix} \hat{Z}_B \cdot \hat{X}_A \\ \hat{Z}_B \cdot \hat{Y}_A \\ \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
$$

- The posture of {B} relative to {A}

$$
{}^A\mathbf{R}_B = \begin{bmatrix} 0.866 & -0.5 & 0 \\ 0.5 & 0.866 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

- The result of swapping the front and back vectors remains unchanged

Means the Projection of X_A on {B}

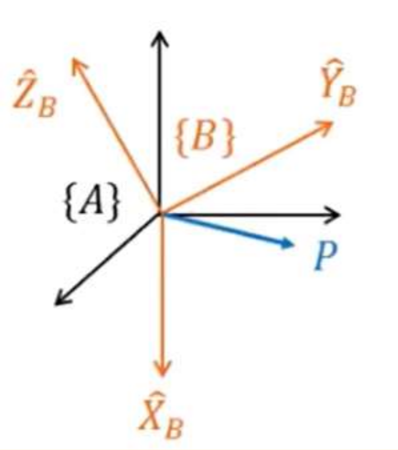
$$
{}^{A}\mathbf{R}_{B} = \begin{bmatrix} \boxed{\hat{X}_A \cdot \hat{X}_B \quad \hat{X}_A \cdot \hat{Y}_B \quad \hat{X}_A \cdot \hat{Z}_B} \\ \hat{Y}_A \cdot \hat{X}_B \quad \hat{Y}_A \cdot \hat{Y}_B \quad \hat{Y}_A \cdot \hat{Z}_B \\ \hat{Z}_A \cdot \hat{X}_B \quad \hat{Z}_A \cdot \hat{Y}_B \quad \hat{Z}_A \cdot \hat{Z}_B \end{bmatrix} = \begin{bmatrix} - & {}^{B}\hat{X}_A{}^{T} & - \\ - & {}^{B}\hat{Y}_A{}^{T} & - \\ - & {}^{B}\hat{Z}_A{}^{T} & - \end{bmatrix}
$$

$$
= \begin{bmatrix} | & | & | \\ {}^{B}\hat{X}_A & {}^{B}\hat{Y}_A & {}^{B}\hat{Z}_A \\ | & | & | \end{bmatrix}^{T} = {}^{B}\mathbf{R}_{A}^{T}
$$

- Second, it can convert coordinates between vectors

$$^{B}P = {}^{B}P_x\hat{X}_B + {}^{B}P_y\hat{Y}_B + {}^{B}P_z\hat{Z}_B$$

$$^{A}P = \boxed{{}^{A}P_x}\hat{X}_A + {}^{A}P_y\hat{Y}_A + {}^{A}P_z\hat{Z}_A$$



$$^{A}P_x = {}^{B}P \cdot \hat{X}_A = \hat{X}_B \cdot \hat{X}_A\,{}^{B}P_x + \hat{Y}_B \cdot \hat{X}_A\,{}^{B}P_y + \hat{Z}_B \cdot \hat{X}_A\,{}^{B}P_z$$

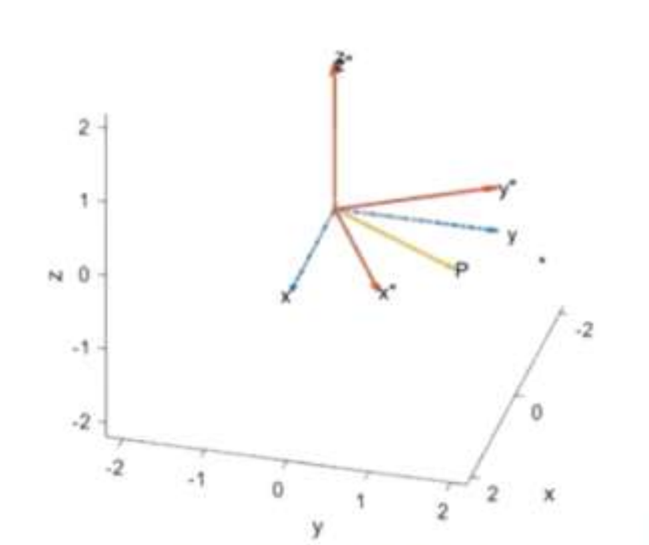
$$^{A}P_y = {}^{B}P \cdot \hat{Y}_A = \hat{X}_B \cdot \hat{Y}_A\,{}^{B}P_x + \hat{Y}_B \cdot \hat{Y}_A\,{}^{B}P_y + \hat{Z}_B \cdot \hat{Y}_A\,{}^{B}P_z$$

$$^{A}P_z = {}^{B}P \cdot \hat{Z}_A = \hat{X}_B \cdot \hat{Z}_A\,{}^{B}P_x + \hat{Y}_B \cdot \hat{Z}_A\,{}^{B}P_y + \hat{Z}_B \cdot \hat{Z}_A\,{}^{B}P_z$$

Projection of vector P from {B} to {A}

$$^{A}P = {}^{A}\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix} {}^{B}\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = {}^{A}_{B}R\,{}^{B}P$$

$$^{B}\mathbf{P} = \begin{bmatrix}1.732 & 1 & 0\end{bmatrix}^{T} \quad ^{A}\mathbf{P} = ?$$



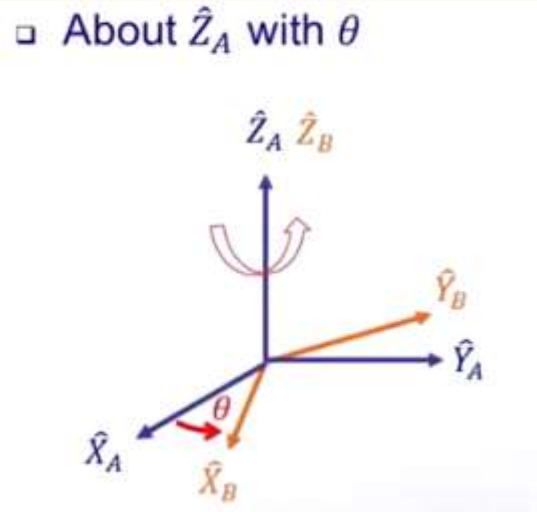
藍虛線: World Frame $\{A\}$

紅實線: Body Frame $\{B\}$

$$^{A}\mathbf{P} = {}^{A}\mathbf{R}_{B} \cdot {}^{B}\mathbf{P}$$

$$^{A}\mathbf{P} = \begin{bmatrix}0.866 & -0.5 & 0\\ 0.5 & 0.866 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1.732\\ 1\\ 0\end{bmatrix} = \begin{bmatrix}1\\ 1.732\\ 0\end{bmatrix}$$

➢ Third, it can further describe the rotational state of the object



Rotation angle (Counter Clockwise)

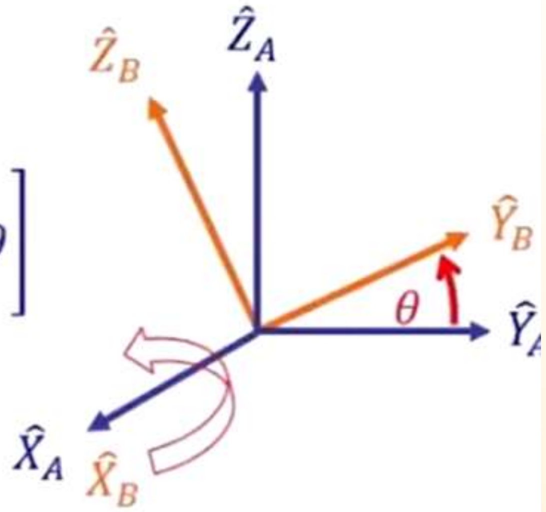
$$R_{\boxed{\hat{z}_A}}(\theta) = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

axis of rotation

$$c\theta = \cos\theta$$
$$s\theta = \sin\theta$$

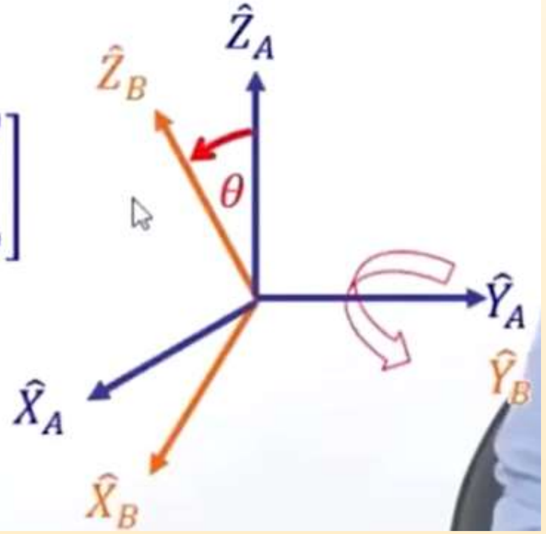
- About $\hat{X}_A$ with θ

$$R_{\hat{X}_A}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\theta & -sin\theta \\ 0 & sin\theta & cos\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\theta & -s\theta \\ 0 & s\theta & c\theta \end{bmatrix}$$



- About $\hat{Y}_A$ with θ

$$R_{\hat{Y}_A}(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} = \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix}$$



Ex:

$${}^{A}P = \begin{bmatrix} 0 & 1 & 1.732 \end{bmatrix}^{T}$$ Rotate 30 degrees to the x-axis, then find ${}^{A}P'$

$$R_{\hat{x}_A}(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 30^\circ & -\sin 30^\circ \\ 0 & \sin 30^\circ & \cos 30^\circ \end{bmatrix} \Rightarrow {}^{A}P' = R_{\hat{x}_A}(\theta)\,{}^{A}P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.866 & -0.5 \\ 0 & 0.5 & 0.866 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1.732 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.866 & -0.5 \\ 0 & 0.5 & 0.866 \end{bmatrix} \qquad = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

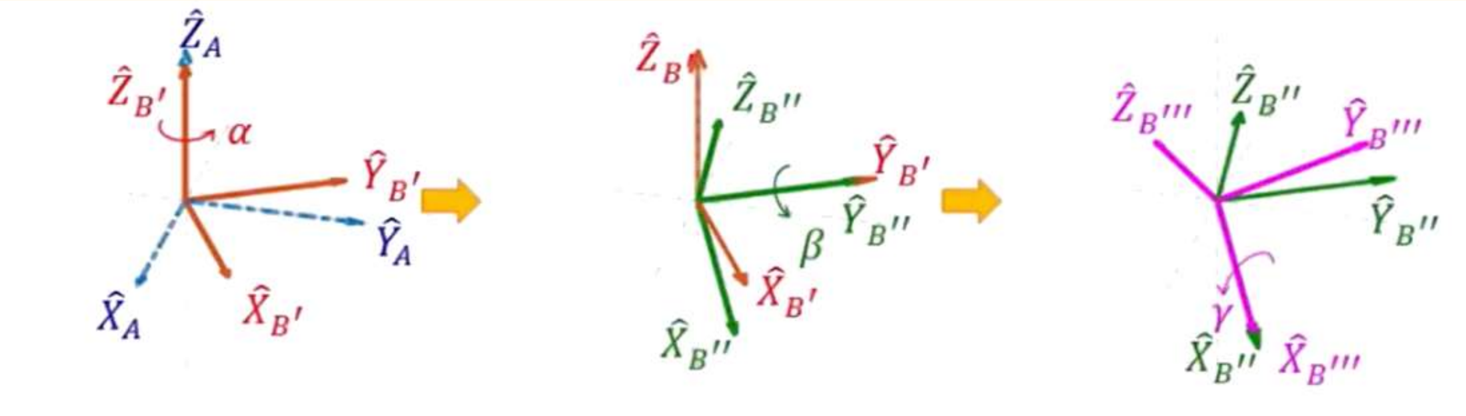
$$\boxed{{}^{A}P' = R(\theta)\,{}^{A}P}$$

Euler Angle

- Any rotation in 3D space can be split becomes a rotation along the three orthogonal coordinate axes of the object itself.

- Matrix multiplication is not commutative, so different rotation orders will produce different results.

➢ First is a Z-Y-X Euler Angle



$$^{A}\mathbf{R}_{BZ'Y'X'}(\alpha,\beta,\gamma) = R_{Z'}(\alpha)R_{Y'}(\beta)R_{X'}(\gamma)$$

$$= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}$$

Ex: What's the difference between "first rotate the x-axis by 60, and then rotate the y-axis by 30" and "first rotate the y-axis by 30, and then rotate the x-axis by 60"

- first rotate the x-axis by 60, and then rotate the y-axis by 30

$$
{}^{A}_{B}\mathbf{R}_{X'Y'Z'}(\gamma,\beta,\alpha) = R'_X(60)R'_Y(30) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 60 & -\sin 60 \\ 0 & \sin 60 & \cos 60 \end{bmatrix} \begin{bmatrix} \cos 30 & 0 & \sin 30 \\ 0 & 1 & 0 \\ -\sin 30 & 0 & \cos 30 \end{bmatrix}
$$

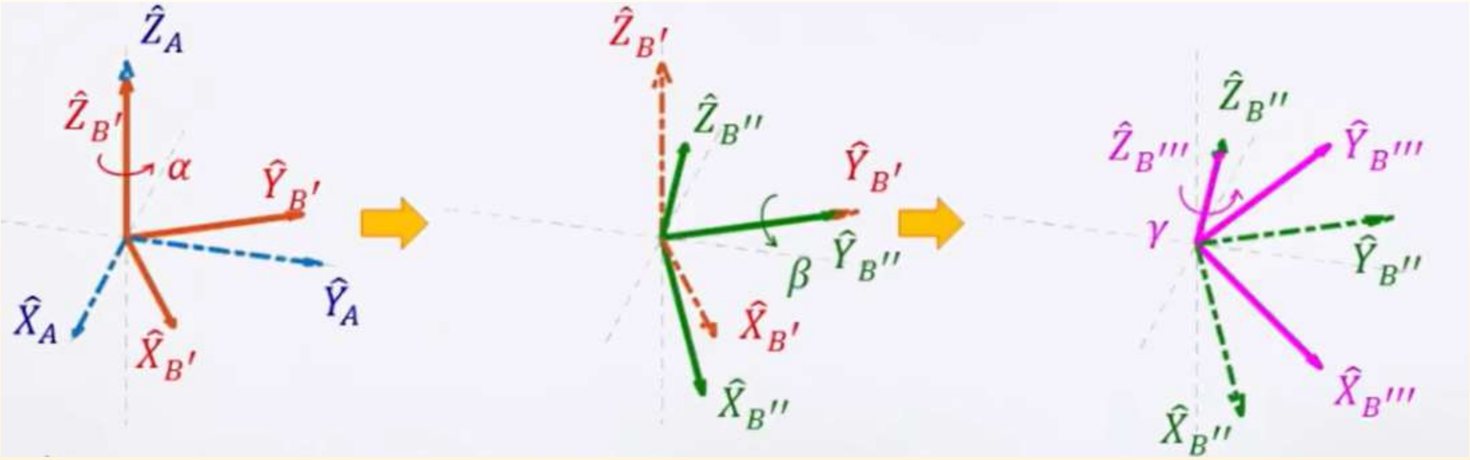
$$
= \begin{bmatrix} 0.866 & 0 & 0.5 \\ 0.433 & 0.5 & -0.75 \\ -0.25 & 0.866 & 0.433 \end{bmatrix}
$$

- first rotate the y-axis by 30, and then rotate the x-axis by 60

$$
{}^{A}_{B}\mathbf{R}_{X'Y'Z'}(\gamma,\beta,\alpha) = R_{Y'}(30^\circ)R_{X'}(60^\circ) = \begin{bmatrix} \cos 30 & 0 & \sin 30 \\ 0 & 1 & 0 \\ -\sin 30 & 0 & \cos 30 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 60 & -\sin 60 \\ 0 & \sin 60 & \cos 60 \end{bmatrix}
$$

$$
= \begin{bmatrix} 0.866 & 0.433 & 0.25 \\ 0 & 0.5 & -0.866 \\ -0.5 & 0.75 & 0.433 \end{bmatrix}
$$

- Second is a Z-Y-Z Euler Angle, which was used in robotics

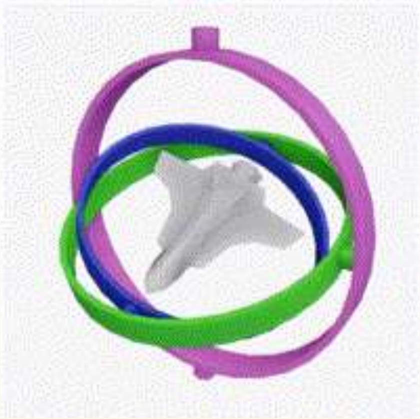


$$
{}^{A}\mathbf{R}_{BZ'Y'Z'}(\alpha,\beta,\gamma) = R_{Z'}(\alpha)R_{Y'}(\beta)R_{Z'}(\gamma)
$$

$$
= \begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}
$$

Gimbal Lock

- However, the Euler Angle is vulnerable to a gimbal lock.



- Gimbal Lock is a phenomenon that occurs when two of the three axes of rotation of a 3D object align, resulting in a loss of one degree of freedom. So the system goes from 3 dimensions to 2 dimensions.

- There are two ways to avoid this problem:

  ·Changing the rotation sequence, if you need to obtain a solution in Euler Angles this is the best way to avoid this problem.

  ·Using **Quaternions**, quaternions do not suffer gimbal lock.

Review Complex Number

- Quaternion is quite similar to Complex Number.

  So first, let us review Complex Number

$$z_1 = a + bi$$

➢ Magnitude: $||z_1|| = \sqrt{a^2 + b^2}$

$$z_2 = c + di$$

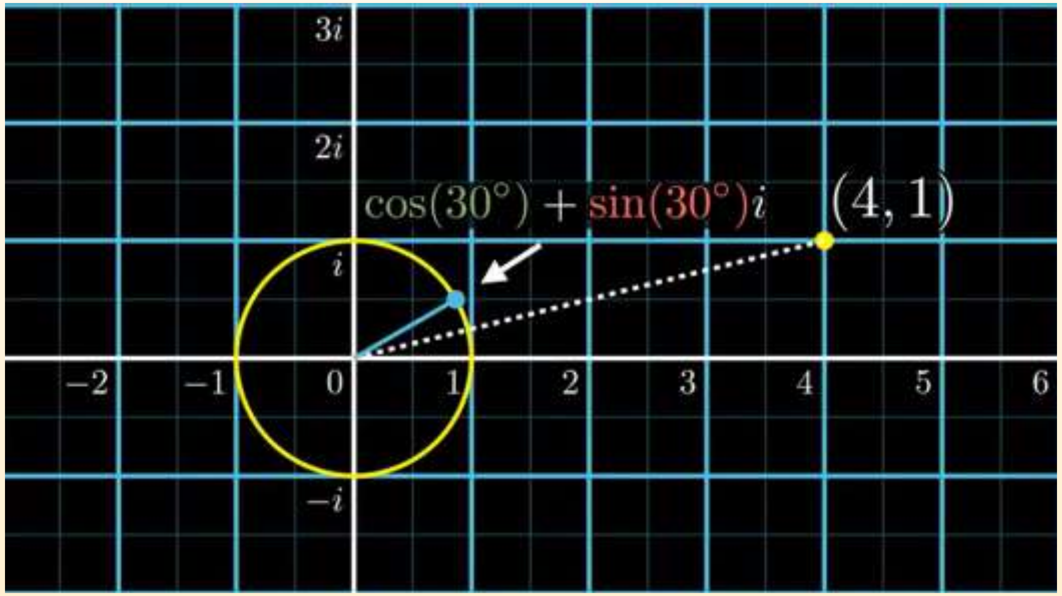
➢ Multiplication:

$$z_1 z_2 = ac + adi + bci + bdi^2 = (ac - bd) + (ad + bc)i$$

$$= \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix}$$

Matrix form of Z1 Vector form of Z2

- 2D Rotation: For example, the vector (4,1) will counterclockwise 30 degrees.



- First, convert (4,1) to 4+1*I

- Then

$$(\cos(30^\circ) + \sin(30^\circ)i)(4 + 1i)$$

$$= (4\cos(30^\circ) - 1\sin(30^\circ)) + (1\cos(30^\circ) + 4\sin(30^\circ))i$$

$$\approx 2.96 + 2.87i$$

Quaternion

Quaternions: $i^2 = j^2 = k^2 = ijk = -1$

$$q = \underbrace{a}_{\text{Scalar part}} + bi + \underbrace{cj + dk}_{\text{Imaginary part or Vector part}}$$

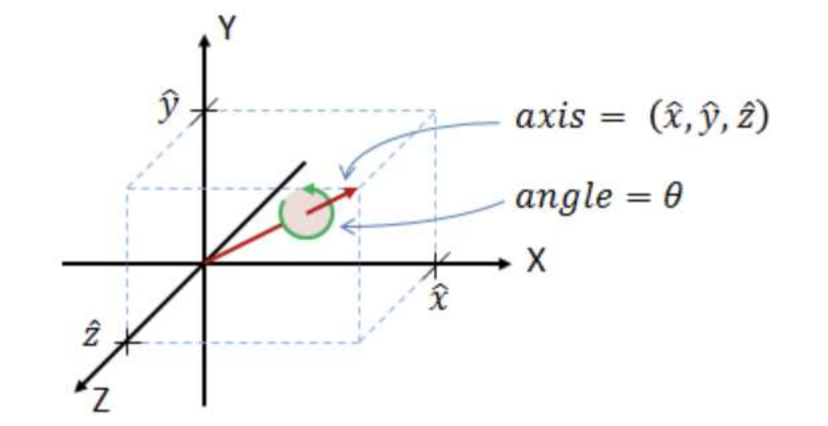
Multiplication:

$$\begin{aligned} q_1 q_2 &= (a + bi + cj + dk)(e + fi + gj + hk) \\ &= (ae - bf - cg - dh)+ \\ &\quad (be + af - dg + ch)i+ \\ &\quad (ce + df + ag - bh)j+ \\ &\quad (de - cf + bg + ah)k. \end{aligned}$$

$$q_1 q_2 = \begin{bmatrix} a & -b & -c & -d \\ b & a & -d & c \\ c & d & a & -b \\ d & -c & b & a \end{bmatrix} \begin{bmatrix} e \\ f \\ g \\ h \end{bmatrix}.$$

Axis-Angle Representation of 3D Rotations

- According to Euler's rotation theorem, any 3D rotation (or sequence of rotations) can be specified using two parameters: a unit vector that defines an axis of rotation; and an angle θ describing the magnitude of the rotation about that axis.



- An axis-angle rotation can therefore be represented by four numbers as

$$(\theta, \hat{x}, \hat{y}, \hat{z})$$

$(\hat{x}, \hat{y}, \hat{z})$ is a unit vector that defines the axis of rotation

θ is the amount of rotation around $(\hat{x}, \hat{y}, \hat{z})$

Convert Axis-Angle to Quaternion

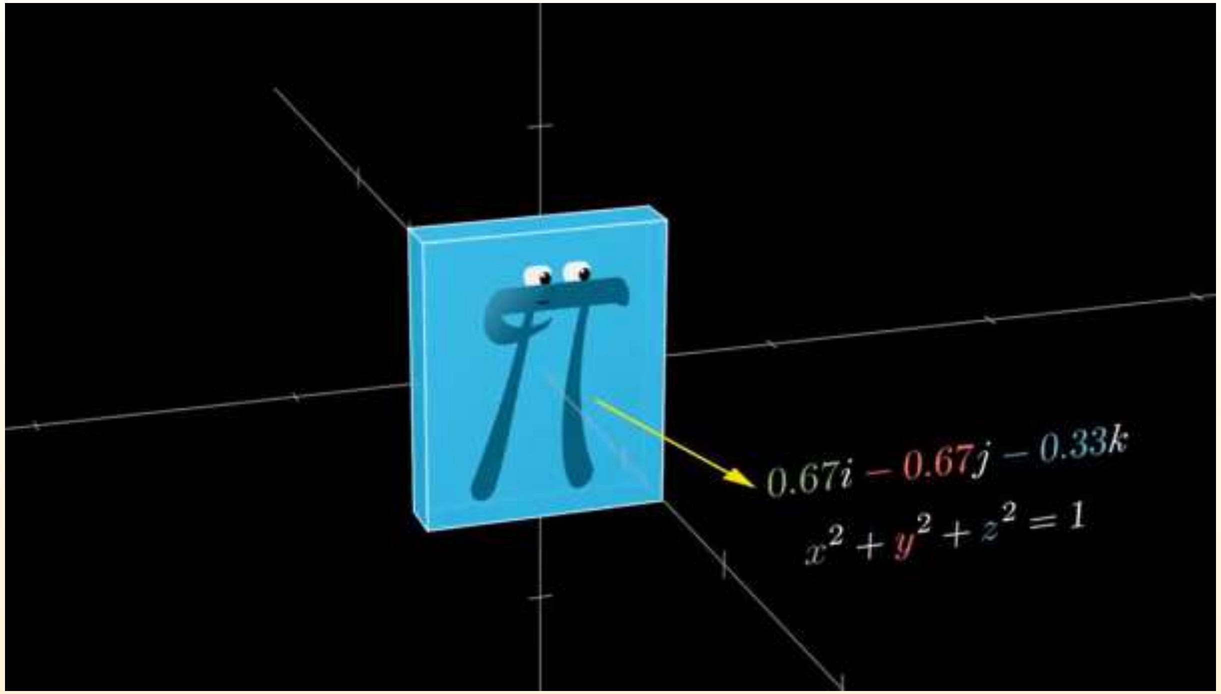
If we know the axis-angle components (θ, $\hat{x}$, $\hat{y}$, $\hat{z}$), we can convert to a rotation quaternion q as follows:

$\boldsymbol{q} = (q_0, q_1, q_2, q_3)$

$$q_0 = \cos\left(\frac{\theta}{2}\right)$$
$$q_1 = \hat{x}\sin\left(\frac{\theta}{2}\right)$$
$$q_2 = \hat{y}\sin\left(\frac{\theta}{2}\right)$$
$$q_3 = \hat{z}\sin\left(\frac{\theta}{2}\right)$$

Using Quaternion to handle

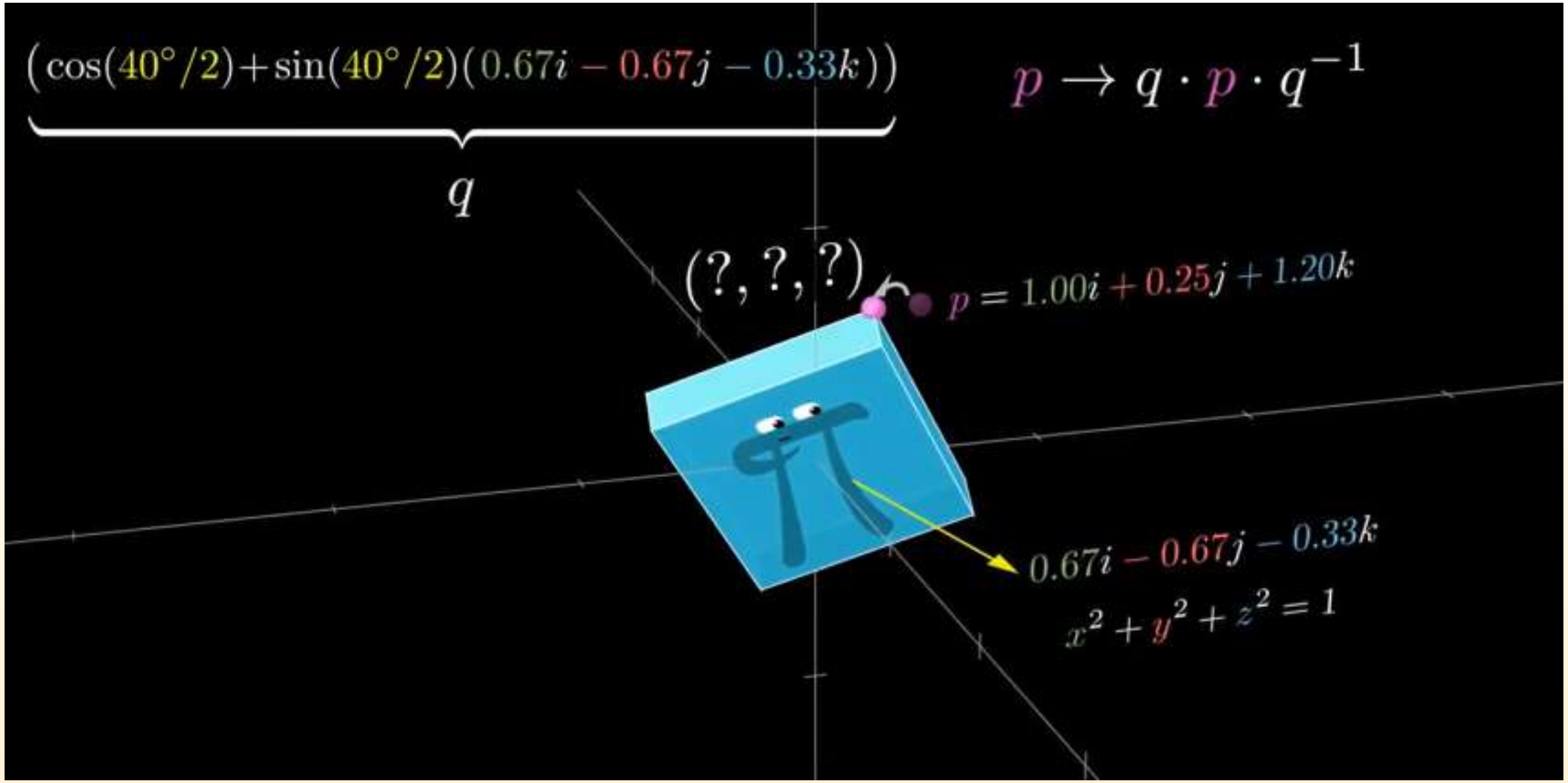
- 3D Rotation



You first define that axis with a unit vector which will write as having i, j, and k components normalized so that the sum of the squares of those components is 1

Second, convert Axis-Angle to Quaternion

$$\underbrace{\left(\cos(40^\circ/2) + \sin(40^\circ/2)(0.67i - 0.67j - 0.33k)\right)}_{q}$$



Rotation of $p = q \cdot p \cdot q^{-1}$

Visualization https://eater.net/quaternions

Why use θ/2

- Rodrigues' rotation formula

$$\mathbf{v}_{\mathrm{rot}} = \mathbf{v}\cos\theta + (\mathbf{k}\times\mathbf{v})\sin\theta + \mathbf{k}(\mathbf{k}\cdot\mathbf{v})(1-\cos\theta).$$

- If we use θ in quaternion rotation, it actually rotates 2θ

$$\left[\cos\theta, \vec{k}\sin\theta\right][0,\vec{v}]\left[\cos\theta, -\vec{k}\sin\theta\right] = \left[-\sin\theta\,\vec{k}\cdot\vec{v}, \cos\theta\,\vec{v} + \sin\theta\,\vec{k}\times\vec{v}\right]\left[\cos\theta, -\vec{k}\sin\theta\right]$$

实数部分：

$$-\sin\theta\cos\theta\,\vec{k}\cdot\vec{v} + \left(\cos\theta\,\vec{v} + \sin\theta\,\vec{k}\times\vec{v}\right)\cdot\left(\vec{k}\sin\theta\right) = 0$$

虚数部分：

$$\begin{aligned}
&\left(\sin\theta\,\vec{k}\cdot\vec{v}\right)\left(\vec{k}\sin\theta\right) + \cos\theta\left(\cos\theta\,\vec{v} + \sin\theta\,\vec{k}\times\vec{v}\right) + \left(\cos\theta\,\vec{v} + \sin\theta\,\vec{k}\times\vec{v}\right)\times\left(-\vec{k}\sin\theta\right)\\
&= \cdots\\
&= (cos^2\theta - sin^2\theta)\vec{v} + 2\cos\theta\sin\theta\,\vec{k}\times\vec{v} + 2sin^2\theta\left(\vec{k}\cdot\vec{v}\right)\vec{k}\\
&= \vec{v}\cos 2\theta + \left(\vec{k}\times\vec{v}\right)\sin 2\theta + \vec{k}(\vec{k}\cdot\vec{v})(1-\cos 2\theta)
\end{aligned}$$

For more detail about how to use Taylor Expansions to get Rodrigues' rotation formula
https://people.eecs.berkeley.edu/~ug/slide/pipeline/assignments/as5/rotation.html

Quaternion Rotation Theorem

Theorem 14: 四元数旋转公式（左手坐标系，右手定则定义正方向）

任意向量 $\mathbf{v}$ 沿着以单位向量定义的旋转轴 $\mathbf{u}$ 逆时针旋转 θ 度之后的 $\mathbf{v}'$ 可以使用四元数乘法来获得．令 $v = [0,\ \mathbf{v}]$，$q = \left[\cos\left(\frac{1}{2}\theta\right),\ \sin\left(\frac{1}{2}\theta\right)\mathbf{u}\right]$，那么：

$$v' = q^* v q = q^{-1} v q$$

Theorem 15: 四元数旋转公式（左手坐标系，左手定则定义正方向）

任意向量 $\mathbf{v}$ 沿着以单位向量定义的旋转轴 $\mathbf{u}$ 顺时针旋转 θ 度之后的 $\mathbf{v}'$ 可以使用四元数乘法来获得．令 $v = [0,\ \mathbf{v}]$，$q = \left[\cos\left(\frac{1}{2}\theta\right),\ \sin\left(\frac{1}{2}\theta\right)\mathbf{u}\right]$，那么：

$$v' = q v q^* = q v q^{-1}$$

Summary

- Single-axis rotations (use Euler)
- Two-axis rotations(use Euler)
- 3-axis rotations(use quaternion)
    1. No gimbal lock/changing axes.
    2. Interpolation is smooth and direct.
    3. Simple to do calculations with

Reference

- Lei_ZM. (2019, May 30). 台大機器人學之運動學 - 林沛群. 哔哩哔哩_bilibili. https://www.bilibili.com/video/BV1v4411H7ez?p=1

- 3Blue1Brown. (2018, October 26). *Quaternions and 3D rotation, explained interactively*. YouTube. https://www.youtube.com/watch?v=zjMuIxRvygQ

- Ena, A. (2022, October 3). *Euler angles, rotations and gimbal lock, brief explanation.* Medium. https://medium.com/@lalesena/euler-angles-rotations-and-gimbal-lock-brief-explanation-de1d4764170

- *Rotation quaternions, and how to use them*. Quaternions. (n.d.). https://danceswithcode.net/engineeringnotes/quaternions/quaternions.html

- Krasjet. (n.d.). *Krasjet/Quaternion: A Brief Introduction to the quaternions and its applications in 3D geometry.* GitHub. https://github.com/Krasjet/quaternion?tab=readme-ov-file