Transformations

Comments	
Dates Taught	@August 14, 2020 → August 21, 2020
Lecture No.	L2 L3 L4
Links of Videos	L2 (Aug 14), L3 (Aug 18), L4 (Aug 21)
Module	Basics
Related to All Questions (Property)	

> Lectures for Aug 14, 18, 21. Please read this note first.

▼ Source for this page

Craig book wherever not explicity mentioned.

▼ Additional Resources

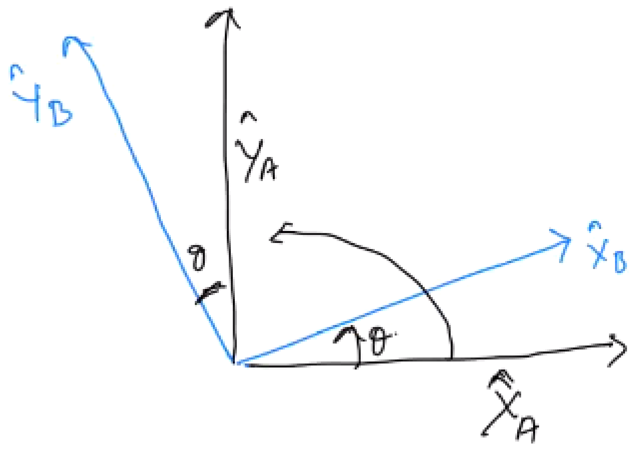
- All resources made available on Teams. link (Use your IIIT account)
- *Main resource is "Introduction to Robotics by Craig".* (highly recommended reading)

---

1. The Tranformation Matrix

1.1 Point, Vector and Coordinate System

Let us first discuss a simple 2D case. If ${}^B\overrightarrow{P}$ is the position of a vector in frame B which has been rotated from the frame A, ${}^A\overrightarrow{P}$ which is the vector in frame A is given by:



$$ {}^AP = R_B^A \; {}^BP $$

$$ \text{where } R_B^A = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} $$

Rotator-transform equivalence (Vector-frame equivalence)

If this section seems confusing, refer to 2.4 section of craig book (Operators).

The standard rotation direction is counter-clockwise. So even if is not mentioned below, you know which direction it is.

1. **Vector (or "operator")**

   If a point is being rotated by θ in the counter-clockwise direction, the coordinates of the new point:

$$ \begin{bmatrix} x \\ y \end{bmatrix}' = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} $$

   So ${}^BP = R_\theta \; {}^AP$ when it is said a vector is undergoing rotation θ. Or in the terminology of Craig, this is a "*rotational operator*" that "operates" on the vector ${}^AP$ and changes into ${}^BP$.

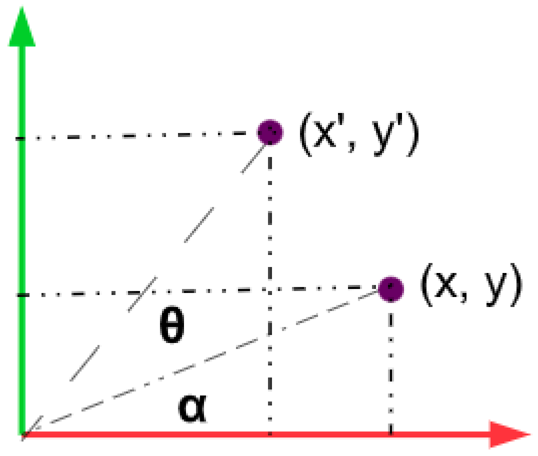
2. **Frame**

   Vector being rotated by θ is equivalent to Frame being rotated by $-\theta$.

   ▼ Why? Think:

   In previous "1. Vector" case, you rotated a vector. You ended up with some coordinate values x', y'. Now, you want to rotate **your frame** such that you have the same coordinate values. (The vector does not move with the rotation of the coordinate system.)

   In other words, **Frame being rotated by** θ is equivalent to Vector being rotated by $-\theta$. So substituting $-\theta$ in the the formula above, the coordinates of the vector in the new frame upon θ frame rotation is given by:

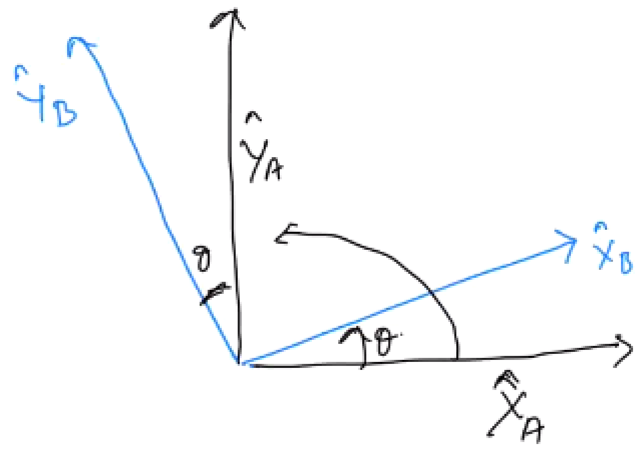


1. Vector

▼ Exercise-1:

Derive the equation using geometry. It's actually easier to derive for "2. Frame" below - One intuitive way would be to think in terms of basis vectors.

$$
\begin{aligned}
\begin{bmatrix} x \\ y \end{bmatrix}' &= \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\
&= R' \begin{bmatrix} x \\ y \end{bmatrix} \\
\implies {}^{B}P &= R' \, {}^{A}P
\end{aligned}
$$



2. Frame: Here in the figure, the frame is being rotated θ.

Here ${}^{A}P = R^{A}_{B} \, {}^{B}P$ but let us derive it from the perspective of "Vector rotation".

Notice what's going on here. In the first case, the vector itself is being rotated (or "being operated on"). In the second case, although the vector is not being rotated, since the frame is being rotated, its coordinates will change. **Those new coordinates of the same vector in the new frame are what we are calling x', y' in the second case, while in the first case, x', y' are the coordinates of the transformed vector in the same frame.**

Now coming to the equivalent R_θ between vector and frame, if you take R' to the other side of the equation (multiply R'^T on both sides), we'd get

$$
{}^{A}P = R_\theta \, {}^{B}P \qquad ({}^{A}P = R^{A}_{B} \, {}^{B}P)
$$

Note that ${}^{A}P, R_\theta, {}^{B}P$ are exactly the same as above, it is just that here we rotate the frame while above, we rotate the vector, hence the same R_θ comes on different sides of the equation.

> **To sum it up, the same rotation matrix R_θ can be thought of as:**
> **>** If I rotate a vector, I want to know where it is going **given my original vector.**
> **>** If I rotate the frame, I want to know how I can get back to vector measured in the original frame **given the vector in rotated frame**.

💡 Quoting Craig:
The rotation matrix that rotates vectors through some rotation, R, (Simply described by R_θ in 1st case) **is the same as the rotation matrix that describes a frame rotated by R relative to the reference frame** (In 2nd case, R_θ is nothing but R^{A}_{B}, i.e. rotation relative to the reference frame)**.** Notice the subtle changes in interpretations.

Let us define our vectors and transformations more generally:

How to describe a vector?

Using a set of bases $(\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3})$, then, the arbitrary vector a has a coordinate under this set of bases:

$$
\mathbf{a} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = a_1\mathbf{e}_1 + a_2\mathbf{e}_2 + a_3\mathbf{e}_3
$$

Most 3D libraries use right-handed (such as OpenGL, 3D Max, etc.) coordinate system, and some libraries use left-handed (such as Unity, Direct3D, etc.). So when you start getting into practical implementation, you need to be very careful.

1.2 Euclidean Transforms

The Euclidean transform consists of rotation and translation. Let's first consider about the rotation. We have an unit-length orthogonal base $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$. After a rotation it becomes $(\mathbf{e}'_1, \mathbf{e}'_2, \mathbf{e}'_3)$. Then, for the same vector $\mathbf{a}$, its coordinates in two coordinate Systems are $[a_1, a_2, a_3]^{\mathrm{T}}$ and $[a_1', a_2', a_3']^{\mathrm{T}}$. Since the vector itself is still the same,

$$[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = [\mathbf{e}_1', \mathbf{e}_2', \mathbf{e}_3'] \begin{bmatrix} a_1' \\ a_2' \\ a_3' \end{bmatrix}$$
$$\text{Multiply by } \begin{bmatrix} \mathbf{e}_1^{\mathrm{T}} \\ \mathbf{e}_2^{\mathrm{T}} \\ \mathbf{e}_3^{\mathrm{T}} \end{bmatrix}$$
$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1^{\mathrm{T}}\mathbf{e}_1' & \mathbf{e}_1^{\mathrm{T}}\mathbf{e}_2' & \mathbf{e}_1^{\mathrm{T}}\mathbf{e}_3' \\ \mathbf{e}_2^{\mathrm{T}}\mathbf{e}_1' & \mathbf{e}_2^{\mathrm{T}}\mathbf{e}_2' & \mathbf{e}_2^{\mathrm{T}}\mathbf{e}_3' \\ \mathbf{e}_3^{\mathrm{T}}\mathbf{e}_1' & \mathbf{e}_3^{\mathrm{T}}\mathbf{e}_2' & \mathbf{e}_3^{\mathrm{T}}\mathbf{e}_3' \end{bmatrix} \begin{bmatrix} a_1' \\ a_2' \\ a_3' \end{bmatrix} \triangleq \mathbf{R}\mathbf{a}'$$

$\mathbf{R}$ is the inner product between two sets of bases. As long as the rotation is the same, this matrix is the same. So It can be said that the matrix $\mathbf{R}$ describes the rotation itself. At the same time, **the components of the matrix are the inner product of the two coordinate system bases**.

We can rewrite the above generic representation in terms of specific coordinate systems A and B as

where

$$^A_BR = \left[\begin{array}{ccc} ^A\hat{X}_B & ^A\hat{Y}_B & ^A\hat{Z}_B \end{array}\right] = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix}$$

Reading between the lines,

$$^A_BR = \left[\begin{array}{ccc} ^A\hat{X}_B & ^A\hat{Y}_B & ^A\hat{Z}_B \end{array}\right] = \begin{bmatrix} ^B\hat{X}_A^T \\ ^B\hat{Y}_A^T \\ ^B\hat{Z}_A^T \end{bmatrix}$$

Therefore, the first row can be seen as $\hat{X}_A$ in B's frame and first column as $\hat{X}_B$ in A's frame. The rows of the matrix are the unit vectors of $\{A\}$ expressed in $\{B\}$ and similarly for $\{B\}$.

Special properties of R

R is an orthogonal matrix with a determinant of 1. Conversely, an orthogonal matrix with a determinant of 1 is also a rotation matrix.

So, you can define a collection of n dimensional rotation matrices as follows:

$$\mathrm{SO}(n) = \left\{\mathbf{R} \in \mathbb{R}^{n \times n} | \mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}, \det(\mathbf{R}) = 1\right\}$$

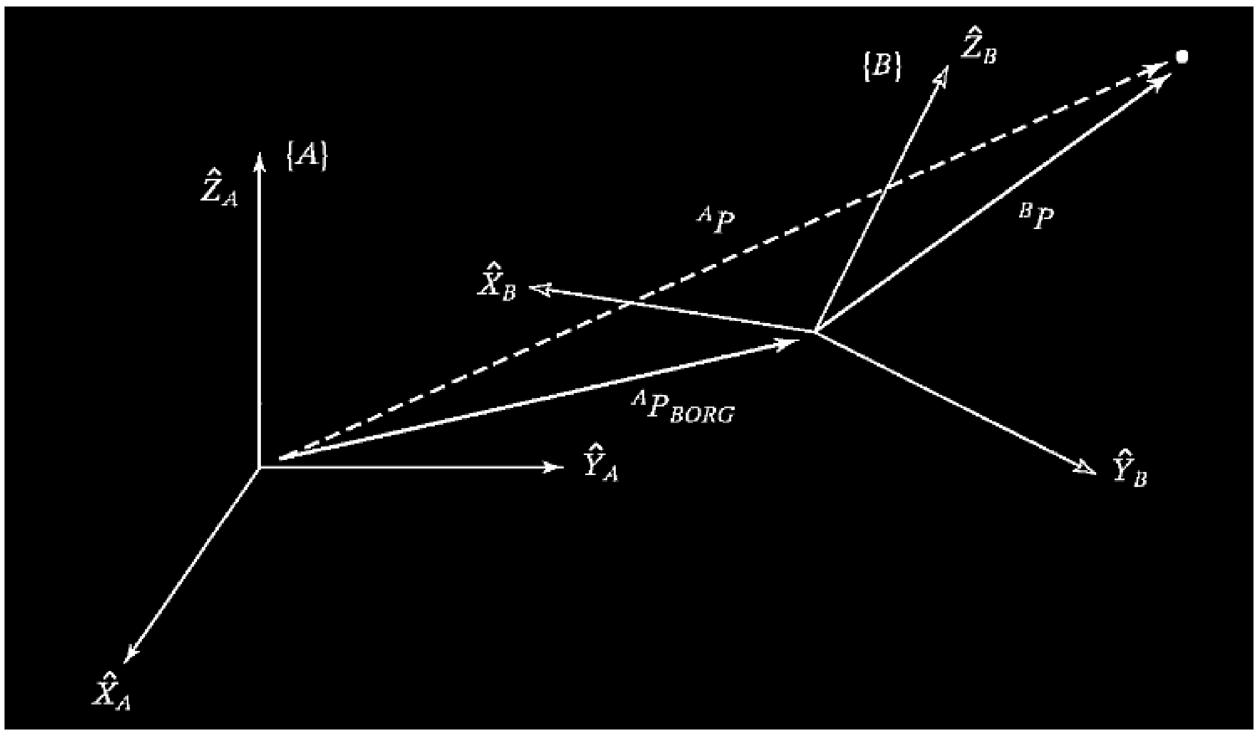
Here, SO stands for "Special Orthogonal" lie group.

▼ lie group?

Lie groups are abstract algebraic constructions used in robotics for representing continuous transformation groups.

It is encouraged but not necessary to go too deep into Lie Algebra and Lie Groups for the scope of this course. This is a great resource if you're interested.

1.3 Transform matrix and homogeneous coordinates



Incorporating the translation of the frame, continuing from here, we get

$$^{A}P = {}^{A}_{B}R\, {}^{B}P + {}^{A}P_{B\ \mathrm{ORG}}$$

Rewriting in a more compact (homogenous) form,

$$\left[\begin{array}{c} {}^{A}P \\ 1 \end{array}\right] = \left[\begin{array}{ccc|c} & {}^{A}_{B}R & & {}^{A}P_{B\ \mathrm{ORG}} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \left[\begin{array}{c} {}^{B}P \\ 1 \end{array}\right]$$

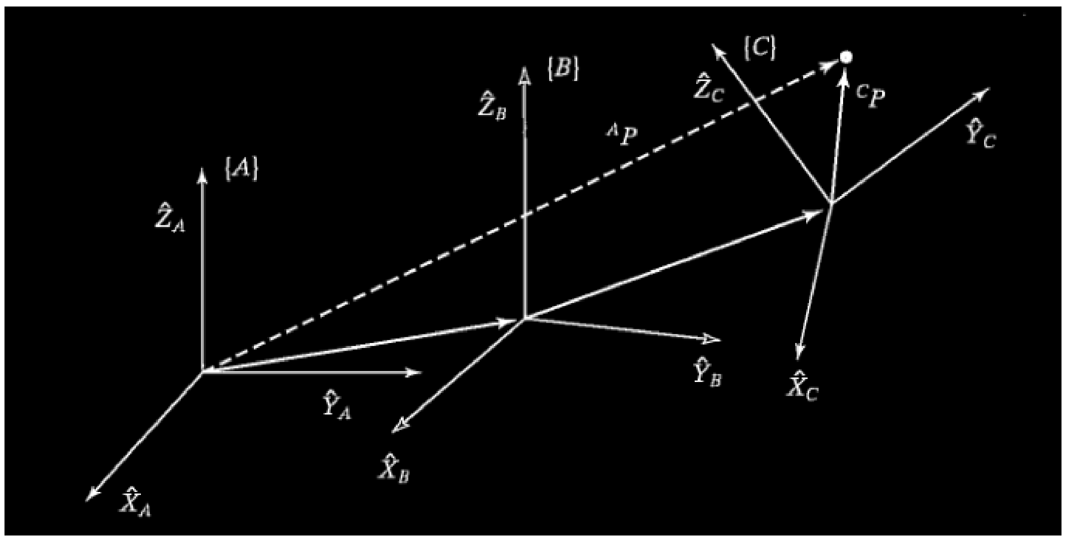
$$^{A}P = {}^{A}_{B}T\, {}^{B}P$$

The T matrix belongs to "Special Euclidean" lie group:

$$\mathrm{SE}(3) = \left\{ \mathbf{T} = \left[\begin{array}{cc} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{array}\right] \in \mathbb{R}^{4\times 4} | \mathbf{R} \in \mathrm{SO}(3), \mathbf{t} \in \mathbb{R}^{3} \right\}$$

$$\mathbf{T}^{-1} = \left[\begin{array}{cc} \mathbf{R}^{\mathrm{T}} & -\mathbf{R}^{\mathrm{T}}\mathbf{t} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{array}\right]$$

1.4 Compound transformations



We have ^{C}P and wish to find ^{A}P and each is known relative to the previous one.

Frame $\{C\}$ is known relative to frame $\{B\}$, and frame $\{B\}$ is known relative to frame $\{A\}$. We can transform ^{C}P into ^{B}P and then, ^{B}P into ^{A}P as

$$^{B}P = {}^{B}_{C}T\ {}^{C}P; \quad {}^{A}P = {}^{A}_{B}T\ {}^{B}P$$

Combining both, we get

$$^{A}P = {}^{A}_{B}T\, {}^{B}_{C}T\ {}^{C}P$$

and

$$ {}^{A}_{C}T = {}^{A}_{B}T \, {}^{B}_{C}T $$

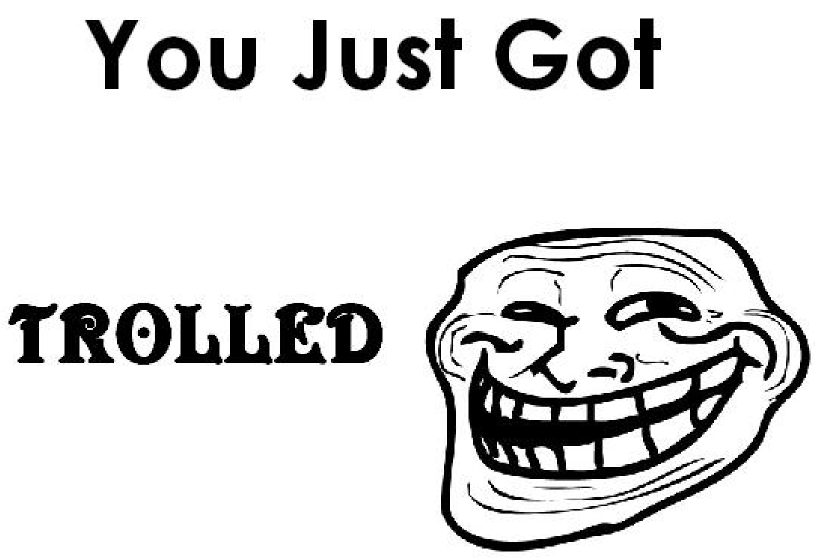
$$ {}^{A}_{C}T = \left[\begin{array}{ccc|c} & {}^{A}_{B}R \, {}^{B}_{C}R & & {}^{A}_{B}R \, {}^{B}P_{C \text{ ORG}} + {}^{A}P_{B \text{ ORG}} \\ \hline 0 & 0 & 0 & 1 \end{array}\right] $$

1.5 Inverting a transform T

Invert the transform ${}^{A}_{B}T$ in terms of ${}^{A}_{B}R$ and ${}^{A}P_{B \text{ ORG}}$.

$$ {}^{B}_{A}T = \left[\begin{array}{ccc|c} & ?? & & ?? \\ \hline 0 & 0 & 0 & 1 \end{array}\right] $$

- ▼ Answer:
    - ▼ Try out yourself first!
        - ▼ Not so easily... Think harder!



> 💡 *Welcome to Mobile Robotics 2020!* 🙂

To un-troll yourself 🤪, go through the Craig book. Good things don't come easy.

---

2. Rotation Vector and Euler Angle

So far we have discussed the Rotation matrix representation. However, there are certain disadvantages: It is not a compact representation and it cannot be visualized easily.

2.1 Rotation Vector

Axis angle

All you really need to describe any rotation is the axis of rotation (say unknown X) and the angle (say unknown Y) by which an object rotates about this axis. This can be measured by a vector whose direction is parallel with the axis of rotation (corresponding to the unknown X) and the length is equal to the magnitude of the angle of rotation (corresponding to the unknown Y), so only a 3D vector is needed. This vector's direction gives X and length gives Y.

If described by a rotation vector, assuming that the rotation axis is a unit length vector $\mathbf{n}$ and the angle is θ, then the vector $\theta\mathbf{n}$ can also describe this rotation.

Rodrigues' Formula: Rotation vector to rotation matrix

$$\mathbf{R} = \cos\theta \mathbf{I} + (1 - \cos\theta)\mathbf{n}\mathbf{n}^{\mathrm{T}} + \sin\theta \mathbf{n}^{\wedge}$$ Proof

Rotation matrix to rotation vector: 1. Finding θ

By taking trace on both sides,

$$\begin{aligned} \operatorname{tr}(\mathbf{R}) &= \cos\theta \operatorname{tr}(\mathbf{I}) + (1 - \cos\theta) \operatorname{tr}\left(\mathbf{n}\mathbf{n}^{\mathrm{T}}\right) + \sin\theta \operatorname{tr}\left(\mathbf{n}^{\wedge}\right) \\ &= 3\cos\theta + (1 - \cos\theta) \\ &= 1 + 2\cos\theta \end{aligned}$$

Hence, θ is given by:

$$\theta = \arccos\left(\frac{\operatorname{tr}(\mathbf{R}) - 1}{2}\right)$$

Rotation matrix to rotation vector: 2. Finding $\mathbf{n}$:

- ▼ $Rn = n$, can anyone explain why this is true? If true, can we make use of it? Clue:
  - ▼ Not so easily... Think before toggling!

    Eigen matrix

2.2 Euler Angles

Two types: Euler angle and Fixed axis.

2.2.1 ZYX Euler angles

> $yaw - pitch - roll$ or ZYX or rpy representation

The **most commonly used Euler angles is the $yaw - pitch - roll$ angles. Equivalent to the rotation of the ZYX axis.**

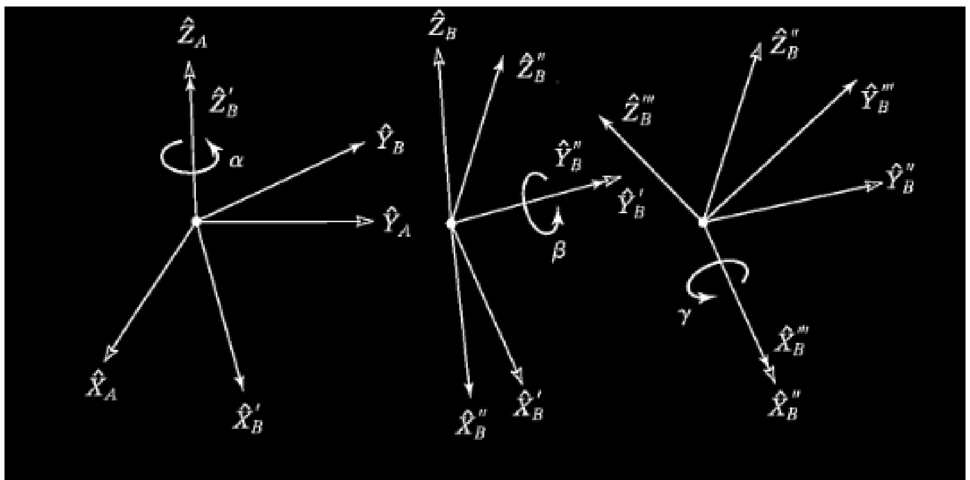


FIGURE 2.18: Z–Y–X Euler angles.

1. Rotate around the Z axis **of the object** to get the yaw angle $\theta_{\text{yaw}} = y = \alpha$
2. Rotate around the Y axis of the object after rotation to get the pitch angle $\theta_{\text{pitch}} = p = \beta$
3. Rotate around the X axis of the object after rotation to get the roll angle $\theta_{\text{roll}} = r = \gamma$

By this way, you can use a three-dimensional vector such as $[r; p; y]^T$ to describe any rotation. **The rotation order of the rpy angle is ZYX.**

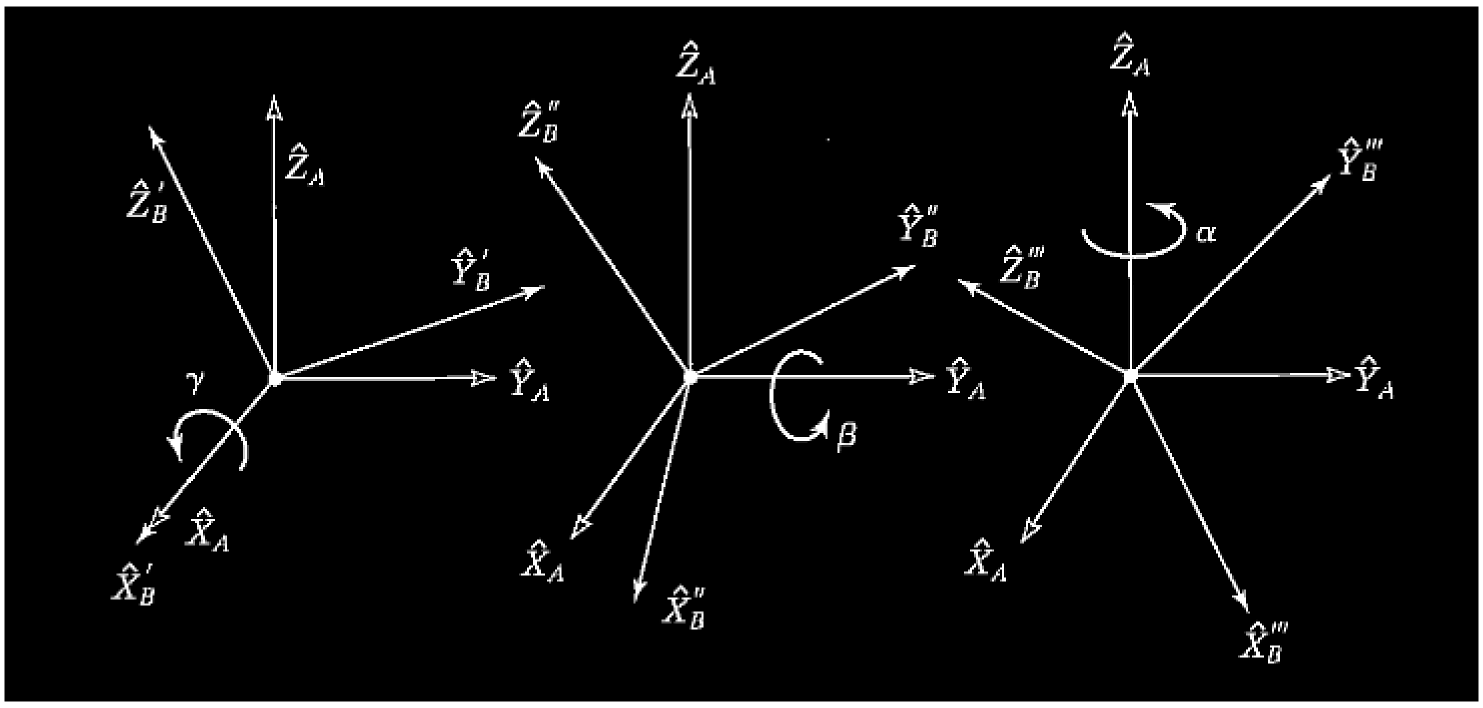
$$\begin{aligned} {}_B^A R_{Z'Y'X'} &= R_Z(\alpha) R_Y(\beta) R_X(\gamma) \\ &= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} \end{aligned}$$

$$
{}^A_B R_{Z'Y'X'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}
$$

2.2.2 Extracting $Z - Y - X$ Euler angles from a rotation matrix

See 2.3.1. The math is exactly the same.

2.3 $X - Y - Z$ Fixed Angles



$$
\begin{aligned}
{}^A_B R_{XYZ}(\gamma,\beta,\alpha) &= R_Z(\alpha) R_Y(\beta) R_X(\gamma) \\
&= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix} \\
{}^A_B R_{XYZ}(\gamma,\beta,\alpha) &= \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}
\end{aligned}
$$

$$
{}^A_B R_{Z'Y'X'}(\alpha,\beta,\gamma) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}
$$

2.3.1 Extracting angles from R matrix

$$
\begin{aligned}
{}^A_B R_{XYZ}(\gamma,\beta,\alpha) &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \\
\beta &= \operatorname{Atan2}\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right) \\
\alpha &= \operatorname{Atan2}\left(r_{21}/c\beta, r_{11}/c\beta\right) \\
\gamma &= \operatorname{Atan2}\left(r_{32}/c\beta, r_{33}/c\beta\right)
\end{aligned}
$$

▼ **Note:** $\operatorname{Atan2}(y,x)$ is a two-argument arc tangent function.

We use this function as $\operatorname{Atan2}(y,x)$ uses the signs of its two arguments to determine which quadrant the angle lies in. Thus, it gives out a unique answer. This makes it better than the regular arctan functions.

Miscellaneous Brainstorming Discussions with students

🏡 Given two vectors, finding rotation matrix using "dot product of basis vectors" method!

▼ Page author(s)

Shubodh Sai