16.485: VNAV - Visual Navigation for Autonomous Vehicles

Lecture 2-3

整理：范浩 （2024秋）

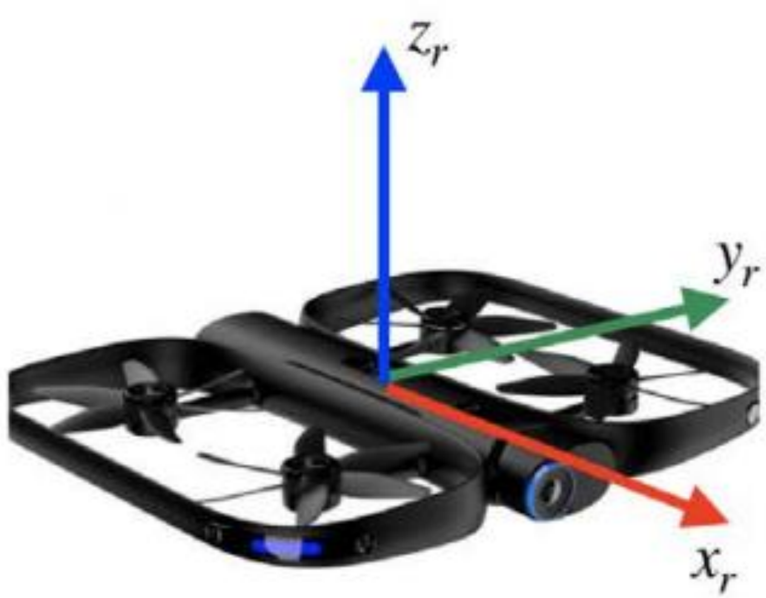
2025.11.3

课程大纲 / Lecture Outline

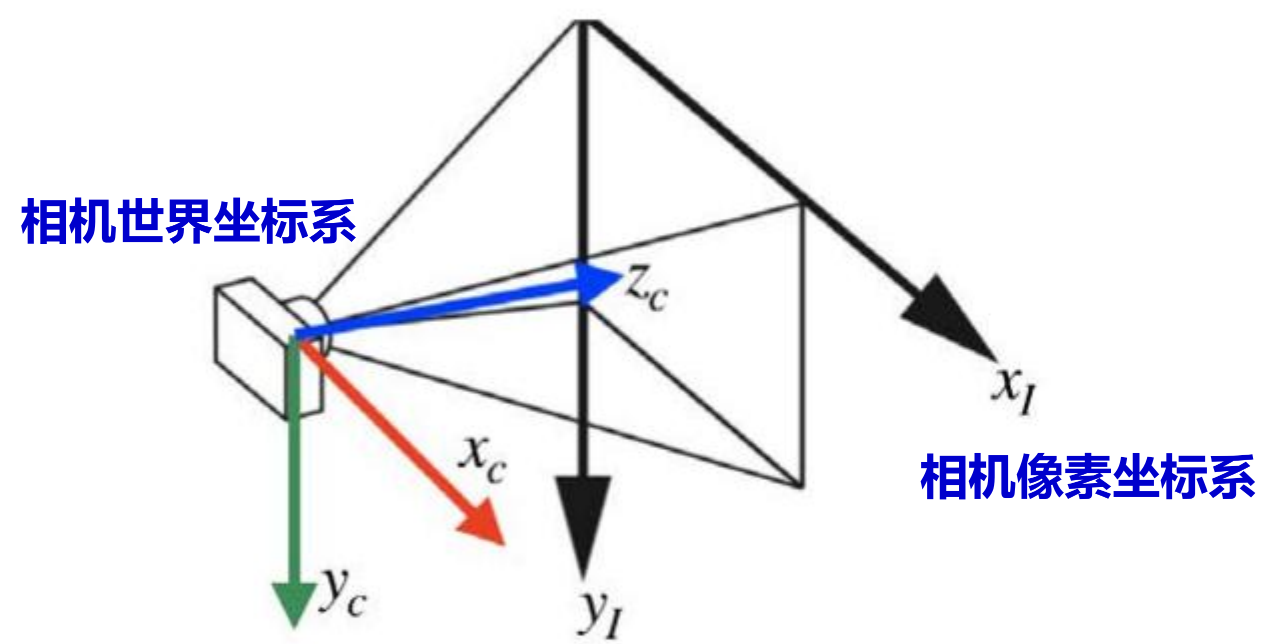
- 坐标系表示 / coordinate frames
- 平移&旋转 / positions and translations
- 姿态表示 / attitude representation
- 位置表示 / pose representation

2.1 坐标系表示 Coordinate Frames

- 在机器人技术中，根据**右手定则**使用**右手坐标系**.
- 机器人本体 (机器人帧 robot frame “r”)；
- 机器人上的传感器 (e.g, 相机帧 camera frame “c”)；
- 自定义的世界坐标系 (world frame “w”)；
- 三维坐标定义 in frame r with axis x_r, y_r, z_r.



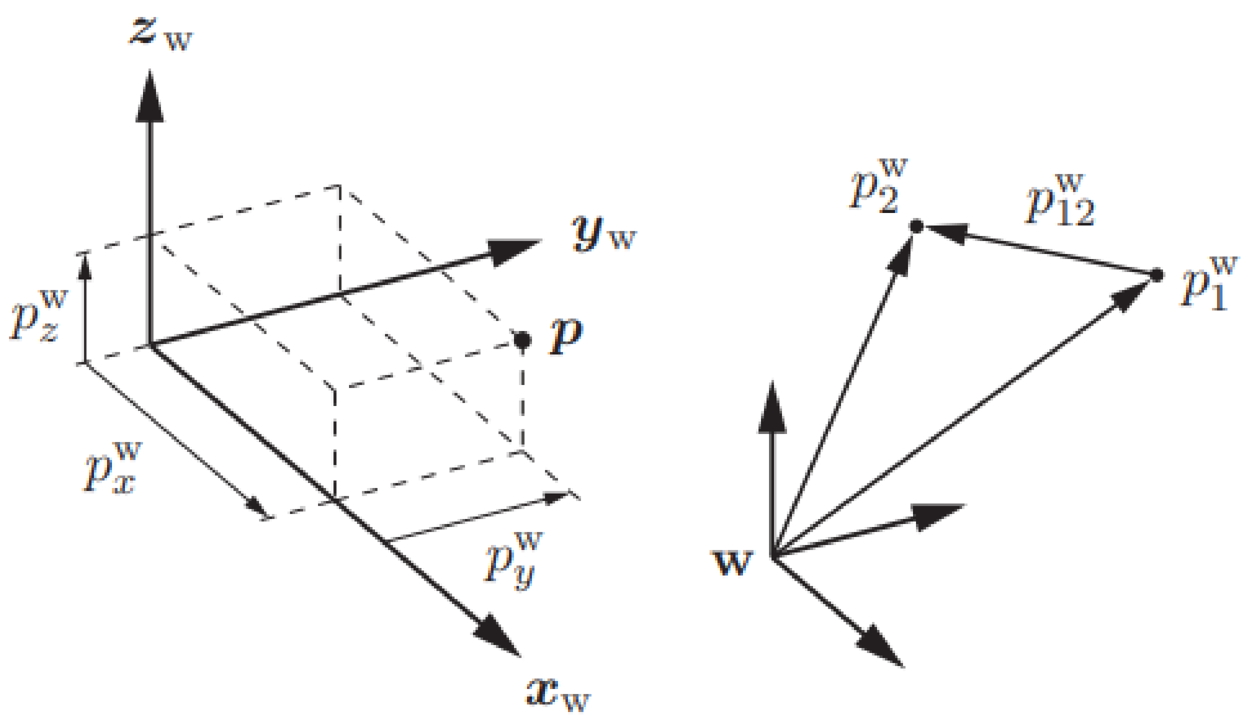
机器人世界坐标系



2.2 点、平移、旋转

Points, positions, and translations

- 使用 3D 矢量表示 3D 点 p 相对于世界帧“w”的位置:



$$\boldsymbol{p}^{\mathrm{w}} = \begin{bmatrix} p_x^{\mathrm{w}} \\ p_y^{\mathrm{w}} \\ p_z^{\mathrm{w}} \end{bmatrix}$$

- **平移的表示**：translation between two points p1 and p2:

$$\boldsymbol{p}_{12}^{\mathrm{w}} = \boldsymbol{p}_2^{\mathrm{w}} - \boldsymbol{p}_1^{\mathrm{w}} \qquad \boldsymbol{p}_2^{\mathrm{w}} = \boldsymbol{p}_1^{\mathrm{w}} + \boldsymbol{p}_{12}^{\mathrm{w}}$$

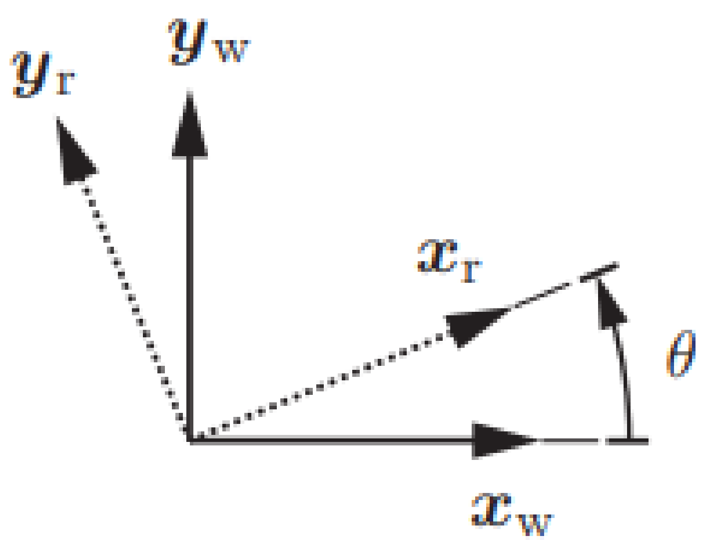
- p_1^w位置 **隐含着** 其相对于坐标系 w 的原点.

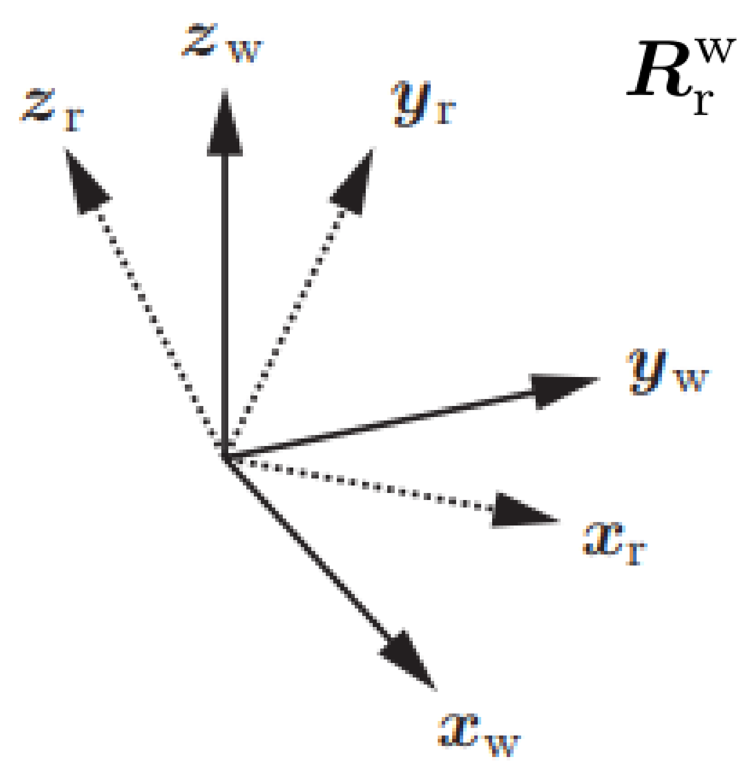
2.3 姿态&旋转 Attitude and rotations

- 上一节中描述的工具允许使用向量来描述给定坐标系中点的位置。
- 然而，在机器人技术中，我们感兴趣的是对可以采取任意位置和方向的物体进行建模。
- 因此，我们在本节中解决的问题是：**我们如何表示一个帧frame（例如 r）相对于另一个帧frame（例如 w ）的方向?**
- 术语：**“方向”**、**“姿态”和“旋转”**可以互换使用，以定义3D 物体方向的直观概念。

2.3.1 旋转矩阵的表示

- **二维旋转矩阵**。轴 x_r 相对于帧frame w 的旋转为:





$$\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin\theta & \cos(\theta) \end{bmatrix}$$

- **3D 旋转矩阵**。我们可以通过在每列中填充世界坐标系 w 中表示的 r 轴的坐标来形成矩阵:

$$\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{r}}^{\mathrm{w}} & \boldsymbol{y}_{\mathrm{r}}^{\mathrm{w}} & \boldsymbol{z}_{\mathrm{r}}^{\mathrm{w}} \end{bmatrix}$$

2.3.1 旋转矩阵的表示

- **旋转矩阵 R_r^W 满足如下性质：**
  - 单位矩阵；
  - 两两向量相互垂直，点乘为0；
  - 矩阵的逆=矩阵的转置。

- *orthogonality*: the axes $\boldsymbol{x}_\mathrm{r}^\mathrm{w}$, $\boldsymbol{y}_\mathrm{r}^\mathrm{w}$, $\boldsymbol{z}_\mathrm{r}^\mathrm{w}$ have unit length and are orthogonal to each other (independently on the reference frame they are expressed in), therefore:

$$\|\boldsymbol{x}_\mathrm{r}^\mathrm{w}\|_2^2 = 1 \qquad \|\boldsymbol{y}_\mathrm{r}^\mathrm{w}\|_2^2 = 1 \qquad \|\boldsymbol{z}_\mathrm{r}^\mathrm{w}\|_2^2 = 1 \qquad \text{(unit length)} \tag{2.7}$$

$$(\boldsymbol{x}_\mathrm{r}^\mathrm{w})^\mathsf{T}\boldsymbol{y}_\mathrm{r}^\mathrm{w} = 0 \qquad (\boldsymbol{x}_\mathrm{r}^\mathrm{w})^\mathsf{T}\boldsymbol{z}_\mathrm{r}^\mathrm{w} = 0 \qquad (\boldsymbol{y}_\mathrm{r}^\mathrm{w})^\mathsf{T}\boldsymbol{z}_\mathrm{r}^\mathrm{w} = 0 \qquad \text{(orthogonal vectors)} \tag{2.8}$$

These relations can be rewritten directly as:

$$(\boldsymbol{R}_\mathrm{r}^\mathrm{w})^\mathsf{T}\boldsymbol{R}_\mathrm{r}^\mathrm{w} = \mathbf{I}_d \qquad \textbf{(orthogonality)} \tag{2.9}$$

where $\mathbf{I}_d$ is the identity matrix of size d (as before, $d = 2$ in 2D problems and $d = 3$ in 3D). A matrix satisfying (2.9) is said to be *orthogonal* and it's easy to see that such a matrix satisfies:

$$(\boldsymbol{R}_\mathrm{r}^\mathrm{w})^{-1} = (\boldsymbol{R}_\mathrm{r}^\mathrm{w})^\mathsf{T} \tag{2.10}$$

2.3.1旋转矩阵的表示

- **旋转矩阵 R_r^W 满足如下性质：**

- *right-handedness*: the 3D axes $\boldsymbol{x}_{\mathrm{r}}^{\mathrm{w}}$, $\boldsymbol{y}_{\mathrm{r}}^{\mathrm{w}}$, $\boldsymbol{z}_{\mathrm{r}}^{\mathrm{w}}$ have to satisfy the right-hand rule, which implies that $\boldsymbol{x}_{\mathrm{r}}^{\mathrm{w}} \times \boldsymbol{y}_{\mathrm{r}}^{\mathrm{w}} = \boldsymbol{z}_{\mathrm{r}}^{\mathrm{w}}$ where $\times$ denotes the cross product between vectors. Since these are unit-length vectors, the following relations hold:

$$\boldsymbol{x}_{\mathrm{r}}^{\mathrm{w}} \times \boldsymbol{y}_{\mathrm{r}}^{\mathrm{w}} = \boldsymbol{z}_{\mathrm{r}}^{\mathrm{w}} \iff (\boldsymbol{z}_{\mathrm{r}}^{\mathrm{w}})^{\mathsf{T}} (\boldsymbol{x}_{\mathrm{r}}^{\mathrm{w}} \times \boldsymbol{y}_{\mathrm{r}}^{\mathrm{w}}) = +1 \iff \det(\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}}) = +1 \tag{2.11}$$

where in the last equality we noticed that the determinant of a 3×3 matrix can be computed as a *triple product* [https://en.wikipedia.org/wiki/Triple_product#Scalar_triple_product] of the columns. In, particular given three vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c} \in \mathbb{R}^3$:

$$\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = \det\left(\begin{bmatrix} \boldsymbol{a} & \boldsymbol{b} & \boldsymbol{c} \end{bmatrix}\right) \qquad \text{and} \tag{2.12}$$

$$\boldsymbol{a} \cdot (\boldsymbol{b} \times \boldsymbol{c}) = -\boldsymbol{c} \cdot (\boldsymbol{b} \times \boldsymbol{a}) = \boldsymbol{c} \cdot (\boldsymbol{a} \times \boldsymbol{b}) \tag{2.13}$$

Hence a rotation matrix $\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}}$ has to satisfy:

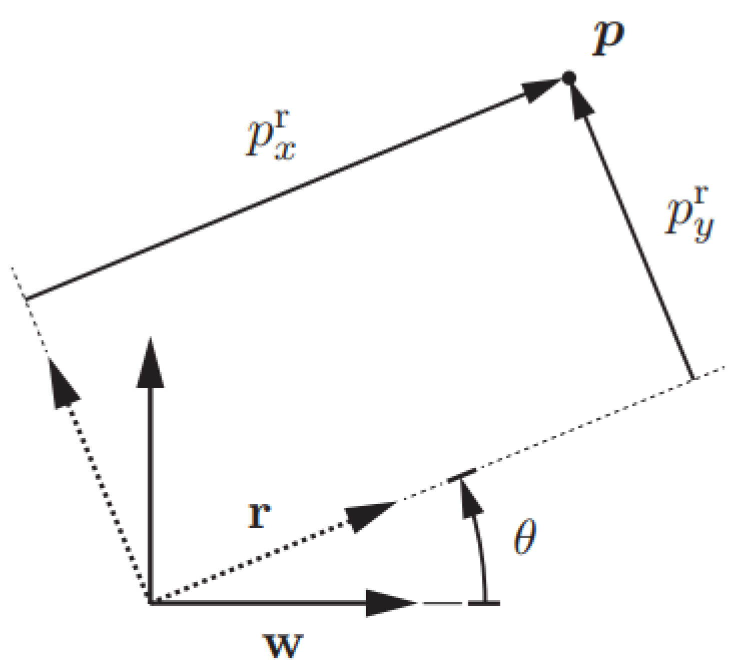
$$\det(\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}}) = +1 \qquad \textbf{(determinant +1)} \tag{2.14}$$

旋转在坐标系帧中表达

- 让我们假设我们得到帧frame r 中一个点的坐标，即 p ， 以及 r 相对于框架 w 的**姿态**，即 R_r^W，其中框架 w 和 r 共享同一原点.

$$\boldsymbol{p}^{\mathrm{w}} = \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{p}^{\mathrm{r}}$$

$$\boldsymbol{p}^{\mathrm{w}} = \boldsymbol{x}_{\mathrm{r}}^{\mathrm{w}}\, p_x^{\mathrm{r}} + \boldsymbol{y}_{\mathrm{r}}^{\mathrm{w}}\, p_y^{\mathrm{r}} + \boldsymbol{z}_{\mathrm{r}}^{\mathrm{w}}\, p_z^{\mathrm{r}} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{r}}^{\mathrm{w}} & \boldsymbol{y}_{\mathrm{r}}^{\mathrm{w}} & \boldsymbol{z}_{\mathrm{r}}^{\mathrm{w}} \end{bmatrix} \begin{bmatrix} p_x^{\mathrm{r}} \\ p_y^{\mathrm{r}} \\ p_z^{\mathrm{r}} \end{bmatrix} = \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{p}^{\mathrm{r}}$$



旋转组成

- 假设赋予一个帧frame r 相对于一个帧frame w 的姿态，即 R_r^W，以及一个框架 c 相对于一个框架 r（即 R_c^r）的姿态.

$$\boxed{\boldsymbol{R}_{\mathrm{c}}^{\mathrm{w}} = \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \ \boldsymbol{R}_{\mathrm{c}}^{\mathrm{r}}}$$

Proof.

$$\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{R}_{\mathrm{c}}^{\mathrm{r}} = \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \begin{bmatrix} \boldsymbol{x}_{\mathrm{c}}^{\mathrm{r}} & \boldsymbol{y}_{\mathrm{c}}^{\mathrm{r}} & \boldsymbol{z}_{\mathrm{c}}^{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{x}_{\mathrm{c}}^{\mathrm{r}} & \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{y}_{\mathrm{c}}^{\mathrm{r}} & \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{z}_{\mathrm{c}}^{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{\mathrm{c}}^{\mathrm{w}} & \boldsymbol{y}_{\mathrm{c}}^{\mathrm{w}} & \boldsymbol{z}_{\mathrm{c}}^{\mathrm{w}} \end{bmatrix} = \boldsymbol{R}_{\mathrm{c}}^{\mathrm{w}}$$

$$\boxed{\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \ \boldsymbol{R}_{\mathrm{c}}^{\mathrm{r}} \neq \boldsymbol{R}_{\mathrm{c}}^{\mathrm{r}} \ \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}}}$$

旋转矩阵的逆

- How can we compute the attitude of w with respect to frame r ?
- 如何计算帧w相对于帧r的姿态呢?

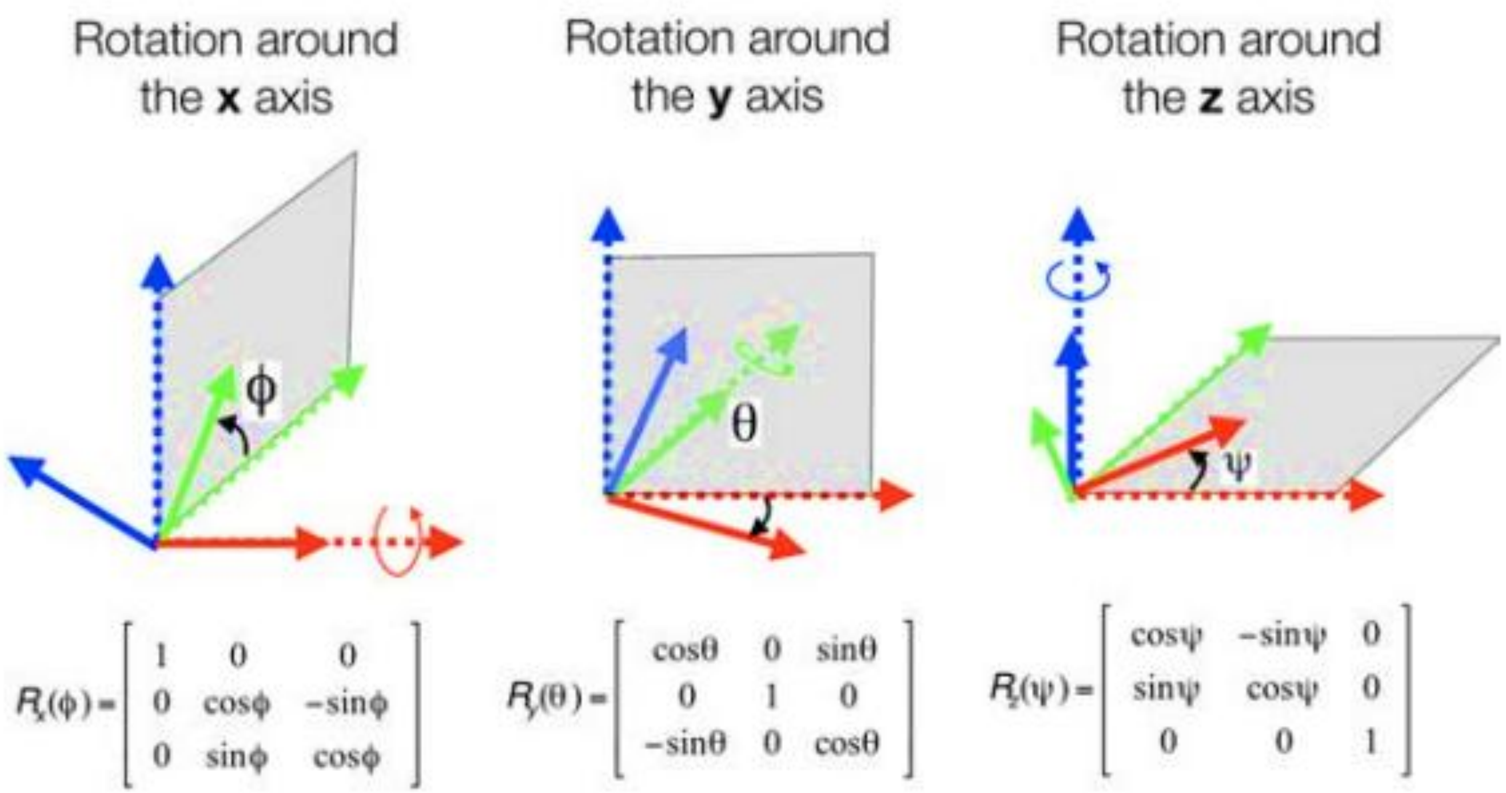
世界帧w 的 坐标 变换到 帧r

$$\boldsymbol{R}_{\mathrm{w}}^{\mathrm{r}} = (\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}})^{-1} = (\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}})^{\mathsf{T}}$$

$$\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{R}_{\mathrm{w}}^{\mathrm{r}} = \boldsymbol{R}_{\mathrm{w}}^{\mathrm{w}} = \mathbf{I}_d \implies \boldsymbol{R}_{\mathrm{w}}^{\mathrm{r}} = (\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}})^{-1} \mathbf{I}_d = (\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}})^{-1}$$

2.3.2 基本旋转和欧拉角表示

Theorem 1 (Euler's rotation theorem, 1775). *Any rotation can be written as the product of no more than three elementary rotations (no two consecutive rotations along the same axis), where the elementary rotations are defined as follows:* **三维旋转可以分解为三个角度的旋转**



A particularly popular choice of Euler angles adopt the following order:

$$\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} = \boldsymbol{R}_z(\gamma)\,\boldsymbol{R}_y(\beta)\,\boldsymbol{R}_x(\alpha)$$

where γ is called the *yaw* angle, β is called the *pitch* angle, and α is called the *roll* angle.

Yaw 航向角（绕Y轴）； Pitch 俯仰角（绕X轴）； roll 横滚角（绕Z轴）

2.4 位姿和刚体变换

- 特别是，如果我们将 t_r^W 称为 r 相对于 w 的原点位置，将 R_r^W 称为 r 相对于 w 的姿态，那么该对:

$$\left(\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}}, \boldsymbol{t}_{\mathrm{r}}^{\mathrm{w}}\right)$$

Pose 的定义很特别!

- 姿势完全由 6 个参数定义（3 个用于平移，3 个用于姿态）。
- 将定义姿势的旋转和平移组装成合适的矩阵实际上很方便:

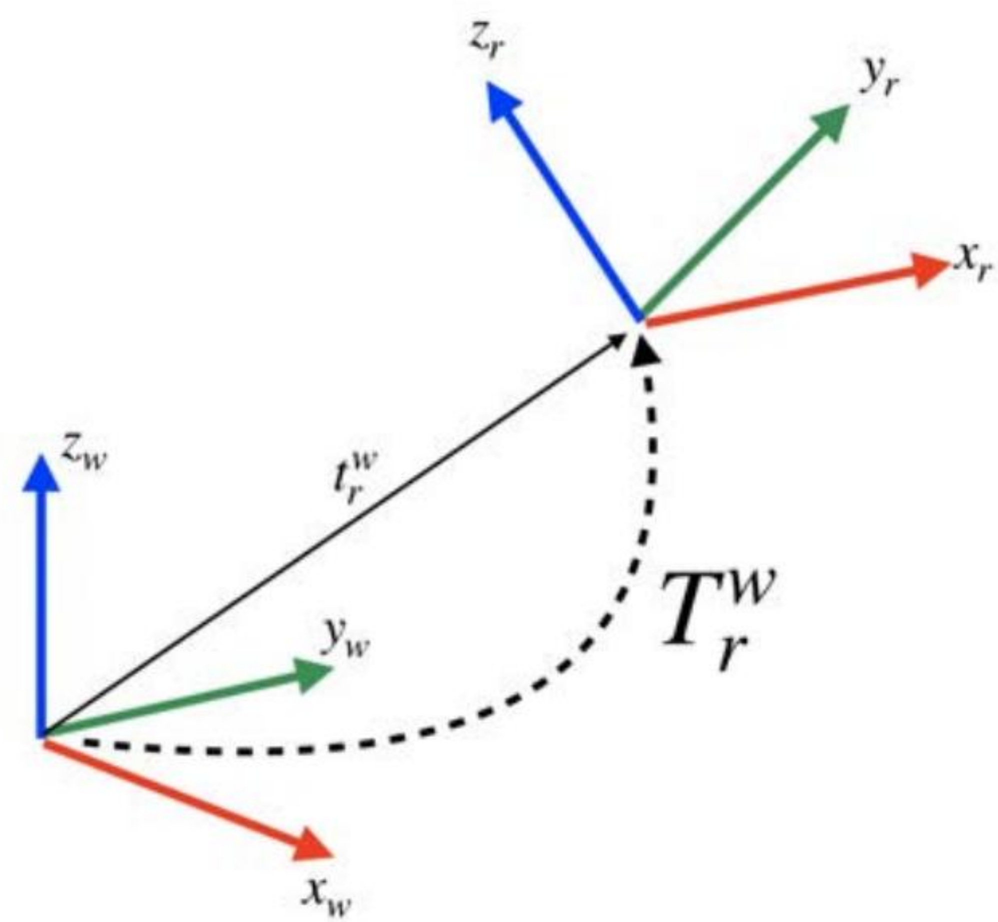
相机帧r 的 坐标 变换到 世界帧w

$$\boldsymbol{T}_{\mathrm{r}}^{\mathrm{w}}=\left[\begin{array}{cc}\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} & \boldsymbol{t}_{\mathrm{r}}^{\mathrm{w}} \\ \mathbf{0}_d^{\top} & 1\end{array}\right]$$

刚体变换

- 假设我们得到一个点 p 的坐标 pr，以参考系 r 表示，并且我们知道参考系 r 相对于系 w 的相对**位姿**，即 T_r^W。
- 那么点 p 相对于帧 w 的位置由下式给出:

$$\boldsymbol{p}^{\mathrm{w}} = \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{p}^{\mathrm{r}} + \boldsymbol{t}_{\mathrm{r}}^{\mathrm{w}}$$



齐次坐标表示:

$$\tilde{\boldsymbol{p}}^{\mathrm{w}} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} & \boldsymbol{t}_{\mathrm{r}}^{\mathrm{w}} \\ \mathbf{0}_d^{\top} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{p}^{\mathrm{r}} \\ 1 \end{bmatrix} = \boldsymbol{T}_{\mathrm{r}}^{\mathrm{w}} \tilde{\boldsymbol{p}}^{\mathrm{r}}$$

位姿组合、位姿的逆

- **位姿组合 Pose composition.**

Similarly to the rotation case, we can compose poses by matrix multiplication:

$$\boldsymbol{T}_{\mathrm{c}}^{\mathrm{w}} = \boldsymbol{T}_{\mathrm{r}}^{\mathrm{w}}\, \boldsymbol{T}_{\mathrm{c}}^{\mathrm{r}}$$

Proof.

$$\boldsymbol{T}_{\mathrm{r}}^{\mathrm{w}}\, \boldsymbol{T}_{\mathrm{c}}^{\mathrm{r}} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} & \boldsymbol{t}_{\mathrm{r}}^{\mathrm{w}} \\ \mathbf{0}_d^{\mathsf{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R}_{\mathrm{c}}^{\mathrm{r}} & \boldsymbol{t}_{\mathrm{c}}^{\mathrm{r}} \\ \mathbf{0}_d^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{R}_{\mathrm{c}}^{\mathrm{r}} & \boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}} \boldsymbol{t}_{\mathrm{c}}^{\mathrm{r}} \\ \mathbf{0}_d^{\mathsf{T}} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_{\mathrm{c}}^{\mathrm{w}} & \boldsymbol{t}_{\mathrm{c}}^{\mathrm{w}} \\ \mathbf{0}_d^{\mathsf{T}} & 1 \end{bmatrix} = \boldsymbol{T}_{\mathrm{c}}^{\mathrm{w}}$$

- **位姿的逆 Inverse of a pose.**

It is possible to show that $\boldsymbol{T}_{\mathrm{w}}^{\mathrm{r}}$ can be computed as follows:

$$\boldsymbol{T}_{\mathrm{w}}^{\mathrm{r}} = (\boldsymbol{T}_{\mathrm{r}}^{\mathrm{w}})^{-1} = \begin{bmatrix} (\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}})^{\mathsf{T}} & -(\boldsymbol{R}_{\mathrm{r}}^{\mathrm{w}})^{\mathsf{T}} \boldsymbol{t}_{\mathrm{r}}^{\mathrm{w}} \\ \mathbf{0}_3^{\mathsf{T}} & 1 \end{bmatrix}$$

References

[1] W.G. Breckenridge. Quaternions - proposed standard conventions. In *JPL, Tech. Rep. INTEROFFICE MEMORANDUM IOM 343-79-1199*, 1999.

[2] E.J. Lefferts, F.L. Markley, and M. D. Shuster. Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429, 1982.

[3] M.D. Shuster. A survey of attitude representations. *Journal of the Astronautical Sciences*, 41(4):439–517, 1993.

[4] J. Stuelpnagel. On the Parametrization of the Three-Dimensional Rotation Group. *SIAM Review*, 6(4):422–430, 1964.

[5] N. Trawny and S.I. Roumeliotis. Indirect kalman filter for 3D attitude estimation. *Mars Lab, Technical Report Number 2005-002, Rev. 57*, 2005.

Experimental course

Lab 1 at 3pm today in C108

Thank you for the attention!