

第一部分

基础知识

第2章 位置与姿态描述

第3章 时间与运动

第2章 位置与姿态描述

机器人和计算机视觉中的一个基本要求是能够表示物体在环境中的位置和方向。这些物体包括机器人、摄像机、工件、障碍物和路径。

空间中的点是数学中一个熟悉的概念，它可以被描述为一个坐标向量，也被称为一个约束向量，如图 2.1(a) 所示。向量表示点相对于某个参考坐标系的位移。一个坐标系或笛卡儿坐标系，是由一组正交轴构成的，这些轴相交于一个被称为原点的点。

更多时候我们需要考虑组成物体的一组点。我们认为物体是刚性的，构成它的点相对于物体坐标系保持固定的相对位置，如图 2.1(b) 所示。然而我们表示物体位置和方向时并不是描述其上单独的点，而是用该物体坐标系的位置和方向来描述。坐标系有自己的标记，比如这里的 $\{B\}$ ，其坐标轴 x_B 和 y_B 采用坐标系的标记作为其下标。

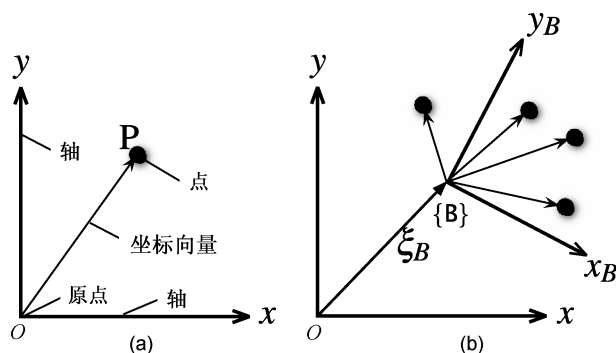
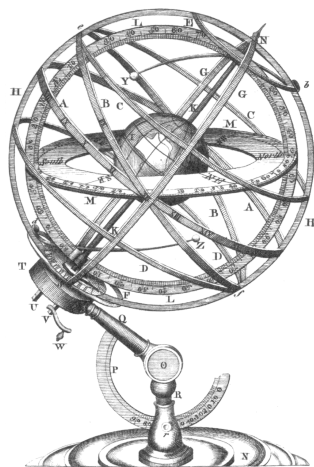


图 2.1 (a) 点 P 由一个相对于绝对坐标系的坐标向量表示。(b) 这些点是用相对于物体坐标系 $\{B\}$ 的向量表示, 该物体坐标系又由一个相对位姿 ξ_B 表示。坐标轴由带有开放式箭头的粗线表示, 向量由一条带有斜箭头的细线表示, 而一条末端为实心箭头的粗线表示一个位姿

坐标系的位置和方向总称为位姿, 图形上表示为一组坐标轴。相对于一个参考坐标系的某个坐标系的相对位姿用符号 ξ 表示。图 2.2 显示了两个坐标系 $\{A\}$ 、 $\{B\}$, 以及 $\{B\}$ 相对于 $\{A\}$ 的相对位姿 ${}^A\xi_B$ 。前面的上标表示参考坐标系, 下标表示被描述的坐标系。我们也可以认为 ${}^A\xi_B$ 描述了一组动作——对 $\{A\}$ 施加平移和旋转使它转化为 $\{B\}$ 。如果没有初始上标, 我们默认位姿的变化是相对于用 O 表示的世界坐标系的。

在图 2.2 中的 P 点可用任何一个坐标系表示。用式子表示为

$${}^A p = {}^A \xi_B \cdot {}^B p \quad (2.1)$$

等式右侧表示从 $\{A\}$ 到 $\{B\}$ 然后到 P 的动作。运算符“ \cdot ”将一个向量转换为一个新的向量, 它们是用一个不同的坐标系来描述的相同点。

相对位姿一个重要的特点是它们可以被合成或组合。以图 2.3 所示情况为例, 如果一个坐标系可以被其他坐标系用相对位姿描述, 那么它们的关系可以记为

$${}^A\xi_C = {}^A\xi_B \oplus {}^B\xi_C$$

用语言可以表述为, $\{C\}$ 相对于 $\{A\}$ 的位姿可由 $\{B\}$ 相对于 $\{A\}$ 的位姿和 $\{C\}$ 相对于 $\{B\}$ 的位姿合成得到。我们利用运算符“ \oplus ”表示相对位姿的合成。

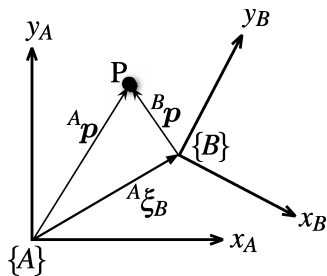


图 2.2 点 P 既可以用相对于坐标系 $\{A\}$ 的坐标向量表示, 又可以用相对于坐标系 $\{B\}$ 的坐标向量表示。坐标系 $\{B\}$ 相对于坐标系 $\{A\}$ 的位姿记作 ${}^A\xi_B$

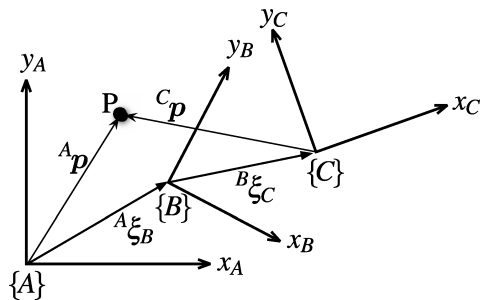


图 2.3 点 P 可以用相对于坐标系 $\{A\}$, $\{B\}$ 或 $\{C\}$ 的坐标向量表示。这些坐标系用相对位姿描述

在这种情况下, 点 P 可以被表示为

$${}^A p = ({}^A\xi_B \oplus {}^B\xi_C) \cdot {}^C p$$

在本章后面, 我们会将这几个抽象的符号“ ξ ”、“ \cdot ”和“ \oplus ”, 转换成可以在 MATLAB 中实现的标准的数学对象和运算符。

在前面的例子中主要讨论了二维坐标系的情况, 能适用于一大类机器人的问题, 特别是在平面世界里作业的移动机器人。对于其他问题, 我们需要用三维坐标系来表示三维世界中的物体, 如飞行机器人、水下机器人的位姿或者机器人手臂夹持的工具末端。

背景材料: 在相对位姿的合成中, 我们可以通过确认运算符 \oplus 两侧的上标和下标字母匹配来检验相对坐标系是否正确。我们可以去掉中间的上、下标字母:

$${}^x\xi_z = \textcircled{x}\xi_x \oplus \textcircled{z}\xi_z$$

只留下左右两头加圈显示的上、下标字母。

图 2.4 中给出了一个更复杂的三维变换的例子, 其中每个实体上都固接了一个三维坐标系, 并表示出了它们之间的相对位姿。固定摄像头从固定的视角观察物体, 并估计物体相对于摄像头本身的位姿。另一个摄像头是不固定的, 它固接到机器人上, 与机器人有固定的相对位姿, 并估计出物体相对于该摄像头的位姿。

另一个表示空间关系的方式是有向图(见附录 J), 如图 2.5 所示。图中的每个节点代表一个位姿, 每条边代表一个相对位姿。从 X 到 Y 的箭头记作 ${}^X\xi_Y$, 表示 Y 相对于 X 的位姿。前面我们提到过可以用运算符 \oplus 来复合合成相对位姿, 这里可以用它写出一些空间关系:

$$\begin{aligned}\xi_F \oplus {}^F\xi_B &= \xi_R \oplus {}^R\xi_C \oplus {}^C\xi_B \\ \xi_F \oplus {}^F\xi_R &= {}^0\xi_R\end{aligned}$$

每个方程表示了图中的一个闭环^①。方程等号两侧的每一边各表示一条网络的通路，即一组按照从头到尾顺序连接的边(箭头线)。等式两边的起始节点和结束节点必须相同。

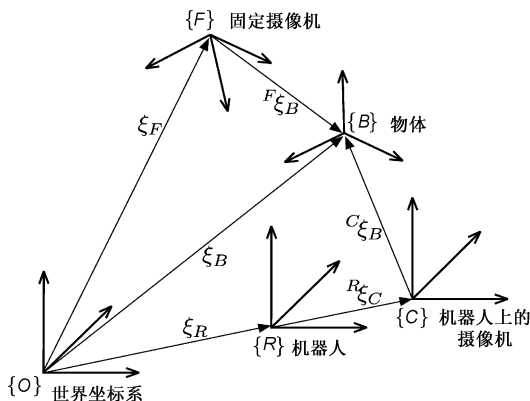


图 2.4 多个三维坐标系及其相对位姿

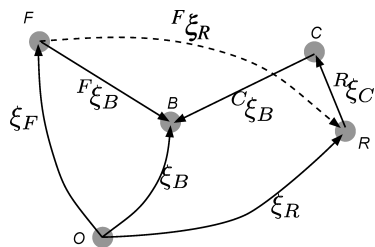
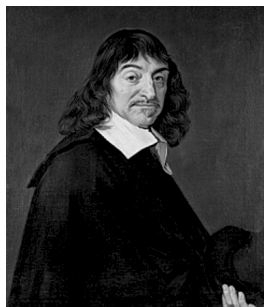


图 2.5 图 2.4 的空间示例被表示为一个有向图

位姿表示的一个非常有用的属性是其代数运算能力。以上面第二个闭环方程为例，它表示机器人的位姿等同于两个相对位姿的合成：从世界坐标系到固定摄像机的相对位姿，以及从固定摄像机到机器人的相对位姿。



背景材料：勒内·笛卡儿 (René Descartes, 1596—1650) 是一个法国哲学家、数学家及兼职雇佣兵。

他因哲学论断“我思，故我在”，或“我思考，所以我存在”，或“我想，因此我在”而著名。他是一个多病的孩子，因而养成了一个伴随终身的习惯：躺在床上思考直到上午。有一个可能是杜撰的故事，讲的是在这样一个早晨，他看见一只苍蝇爬过天花板，这时他意识到可以用它与天花板两条边的距离来描述它的位置。这个坐标系，即笛卡儿坐标系，构成了现代(分析)几何的基础，并影响了现代微积分的发展。在瑞典时，由于女王克里斯汀的邀请，他不得不早上 5 点起床，这打破了他的终身习惯——因此得了肺炎而死。他的遗体后来被运回巴黎，然后又搬动了好几次，现在还有一些关于他的遗体存放位置的争论。他死后，罗马天主教会曾把他的作品列入违禁书籍。

我们可以通过在等式两边同时加上一个逆 ξ_F 来消除 ξ_F ，记作

$$\begin{aligned}\ominus \xi_F \oplus \xi_F \oplus {}^F \xi_R &= \ominus \xi_F \oplus \xi_R \\ {}^F \xi_R &= \ominus \xi_F \oplus \xi_R\end{aligned}$$

其结果就是机器人相对于固定摄像机的位姿。

下面是几个代数运算规则：

$$\begin{aligned}\xi \oplus 0 &= \xi, \quad \xi \ominus 0 = \xi \\ \xi \ominus \xi &= 0, \quad \ominus \xi \oplus \xi = 0\end{aligned}$$

^① 在数学对象术语中位姿组构成了一个群，即一系列对象的集合，它支持一种运算结果仍属于该群的关联二元运算符(复合算子)，一种求逆运算和一种单位元素。在这里，该群是二维或三维的特殊欧几里得群，分别记作 $SE(2)$ 或 $SE(3)$ 。

其中0表示一个零相对位姿。一个位姿可以有逆位姿：

$$\ominus^X \xi_Y = {}^Y \xi_X$$

该逆位姿在图中可表示为从Y到X的一条箭头线。相对位姿可以被复合或合成：

$${}^X \xi_Y \oplus {}^Y \xi_Z = {}^X \xi_Z$$

需要注意的是位姿的代数运算规则不同于一般的代数运算规则，位姿复合运算中各个项的位置不能交换：

$$\xi_1 \oplus \xi_2 \neq \xi_2 \oplus \xi_1$$

$\xi_1 \oplus \xi_2 = 0$ 时才可以互换位置。可以用相对位姿将一个点从一个坐标系中的一个向量转换为另一个坐标系中的另一个向量：

$${}^X p = {}^X \xi_Y \cdot {}^Y p$$

那么，什么是 ξ ？它可以是支持上述代数运算的任何数学对象，并适用于我们身边常见的问题。它还取决于我们讨论的是二维还是三维的问题。在本章接下来的内容中我们将讨论一些对象，其中包括向量以及一些更奇特的数学对象，如齐次变换矩阵、正交旋转矩阵和四元数等。幸运的是，所有这些数学对象都是非常适合 MATLAB 的数学编程环境。

简要回顾一下前面的内容：

1. 一个点用坐标向量来表示，它代表该点在参考坐标系中的位移；
2. 一个刚体可以由其上的一组点代表，该刚体可以用单独一个坐标系描述，并且组成它的点可以用它们在该坐标系中的位移来表示；
3. 一个物体在坐标系中的位置和方向称为它的位姿；
4. 一个相对位姿表示一个坐标系相对于另一个坐标系的位姿，记作代数变量 ξ ；
5. 一个点可以用不同坐标系中的不同坐标向量来描述，向量之间通过坐标系相对位姿来转换，其运算符为“ \cdot ”；
6. 用相对位姿写成的代数表达式是可以进行代数运算的。

本章的剩余部分将讨论在机器人和计算机视觉中常见的各种 ξ 的具体表现。

2.1 二维空间位姿描述

二维世界或平面，是我们在高中学习欧几里得几何时就熟悉的。笛卡儿坐标系，或以 x 轴和 y 轴为正交轴的坐标系，通常绘制成 x 轴水平、 y 轴竖直，两轴的交点称为原点。平行于坐标轴的单位向量用 \hat{x} 和 \hat{y} 表示。一个点用其在 x 轴和 y 轴上的坐标 (x, y) 表示，或者写为有界向量：

$$p = x\hat{x} + y\hat{y} \quad (2.2)$$

在图2.6中的一个坐标系 $\{B\}$ ，我们希望用参照系 $\{A\}$ 来描述它。可以清楚地看到， $\{B\}$ 的原点已被向量 $t = (x, y)$ 所取代，然后逆时针旋转一个角度 θ 。因此，位姿的一个具体表示就是三维向量 ${}^A \xi_B \sim (x, y, \theta)$ ，我们使用符号 \sim 表示这两种表

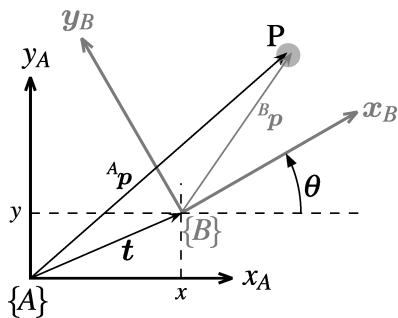


图2.6 两个二维坐标系 $\{A\}$ 、 $\{B\}$ ，以及一个点 P 。 $\{B\}$ 相对于 $\{A\}$ 做旋转和平移

示是等价的。遗憾的是,这种表示方法不方便复合,因为

$$(x_1, y_1, \theta_1) \oplus (x_2, y_2, \theta_2)$$

两边的位姿都是复杂的三角函数。所以,我们将使用一种不同的方法来表示旋转。

该方法是考虑一个任意点 \mathbf{P} 相对于每个坐标系的向量,并确定^A \mathbf{p} 和^B \mathbf{p} 之间的关系。再次回到图 2.6,我们将问题分成两部分:旋转,然后平移。



背景材料: 欧几里得 (Euclid of Alexandria, 约公元前 325—公元前 265) 是被誉为“几何之父”的埃及数学家。在其专著《元素》(*Elements*)中,他从一小组公理中推断出了几何对象和整数的属性。

《元素》可能是在数学史上最成功的一本书,它描述了平面几何原理,是大多数人首次接触几何和公式化证明的基础,也是现代欧几里得几何的基础。欧几里得距离是简单的在一个平面上两个点之间的距离。欧几里得还写了《光学》一书,主要介绍几何视觉和透视。

先只考虑旋转的情况,我们创建一个新坐标系 $\{V\}$,其坐标轴平行于坐标系 $\{A\}$ 的轴,但其原点与坐标系 $\{B\}$ 的原点重合,如图 2.7 所示。根据方程(2.2),我们可以将点 P 用 $\{V\}$ 中定义坐标轴的单位向量表示为

$$\begin{aligned} {}^V\mathbf{p} &= {}^Vx\hat{x}_V + {}^Vy\hat{y}_V \\ &= (\hat{x}_V \ \hat{y}_V) \begin{pmatrix} {}^Vx \\ {}^Vy \end{pmatrix} \end{aligned} \quad (2.3)$$

上式被写作一个行向量和一个列向量的点积。

坐标系 $\{B\}$ 可以用它的两个正交轴表示,这里用两个单位向量代表:

$$\begin{aligned} \hat{x}_B &= \cos\theta\hat{x}_V + \sin\theta\hat{y}_V \\ \hat{y}_B &= -\sin\theta\hat{x}_V + \cos\theta\hat{y}_V \end{aligned}$$

上式用矩阵形式可以分解成:

$$(\hat{x}_B \ \hat{y}_B) = (\hat{x}_V \ \hat{y}_V) \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad (2.4)$$

用方程(2.2)可以在坐标系 $\{B\}$ 中将 \mathbf{P} 点表示为

$${}^B\mathbf{p} = {}^Bx\hat{x}_B + {}^By\hat{y}_B = (\hat{x}_B \ \hat{y}_B) \begin{pmatrix} {}^Bx \\ {}^By \end{pmatrix}$$

代入方程(2.4),得

$${}^B\mathbf{p} = (\hat{x}_V \ \hat{y}_V) \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} {}^Bx \\ {}^By \end{pmatrix} \quad (2.5)$$

现在令方程(2.3)和方程(2.5)各自右侧的系数部分相等,可得

$$\begin{pmatrix} {}^Vx \\ {}^Vy \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} {}^Bx \\ {}^By \end{pmatrix}$$

上式描述了点如何通过坐标系旋转从坐标系 $\{B\}$ 变换到坐标系 $\{V\}$ 。这种类型的矩阵被称为旋转矩阵,记作^V \mathbf{R}_B 。

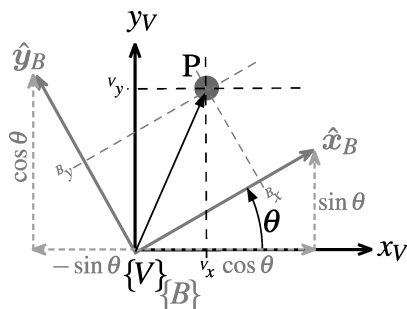


图 2.7 二维平面中旋转后的坐标系。点 \mathbf{P} 可以分别看作在 $\{V\}$ 或 $\{B\}$ 坐标系中的点

$$\begin{pmatrix} {}^V x \\ {}^V y \end{pmatrix} = {}^V R_B \begin{pmatrix} {}^B x \\ {}^B y \end{pmatrix} \quad (2.6)$$

旋转矩阵 ${}^V R_B$ 具有一些特殊的属性。首先，它是正规化的(也称为标准正交)，因为它的每列都是单位向量且相互正交的^①。实际上矩阵的每列都是简单地将 $\{B\}$ 定义在 $\{V\}$ 中的单位向量，因此根据定义它们都是单位长度且正交的。

其次，它的行列式是+1，这意味着 R 属于特殊的二维正交群，或 $R \in SO(2) \subset \mathbb{R}^{2 \times 2}$ 。而且单位行列式还意味着一个向量在变换后的长度是不变的，即 $|\mathbf{p}| = |{}^V \mathbf{p}|$ ， $\forall \theta$ 。

正交矩阵有一种非常方便的属性： $R^{-1} = R^T$ ，即它的逆矩阵和转置矩阵相同。因此，我们可以重新将方程(2.6)整理为

$$\begin{pmatrix} {}^B x \\ {}^B y \end{pmatrix} = ({}^V R_B)^{-1} \begin{pmatrix} {}^V x \\ {}^V y \end{pmatrix} = ({}^V R_B)^T \begin{pmatrix} {}^V x \\ {}^V y \end{pmatrix} = {}^B R_V \begin{pmatrix} {}^V x \\ {}^V y \end{pmatrix}$$

我们注意到，该矩阵的求逆就是将矩阵的上、下标交换位置，并可以得出恒等式 $R(-\theta) = R(\theta)^T$ 。

这里我们观察到一个有趣的事实：我们描述一个旋转的时候，不是用代表旋转角度的一个标量，而是用了一个有4个元素的 2×2 矩阵。但这4个元素并不是独立的，矩阵的每一列都是一个单位的大小，这提供了两个约束，列与列之间还都是正交的，这提供了另一种约束。4个元素加上3个约束，这样还是只剩下1个真正独立的值。旋转矩阵是一个非最小化表示的典型例子，虽然这种表示有一些缺点，诸如需要增加内存等，但它具备的优势更加突出，如可复合性。

描述位姿的第二部分就涉及图2.6所示的各坐标系原点的平移。由于坐标系 $\{V\}$ 和 $\{A\}$ 的轴是平行的，所以可以简单地进行了向量相加：

$$\begin{pmatrix} {}^A x \\ {}^A y \end{pmatrix} = \begin{pmatrix} {}^V x \\ {}^V y \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.7)$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} {}^B x \\ {}^B y \end{pmatrix} + \begin{pmatrix} x \\ y \end{pmatrix} \quad (2.8)$$

$$= \begin{pmatrix} \cos\theta & -\sin\theta & x \\ \sin\theta & \cos\theta & y \end{pmatrix} \begin{pmatrix} {}^B x \\ {}^B y \\ 1 \end{pmatrix} \quad (2.9)$$

或简写为

$$\begin{pmatrix} {}^A x \\ {}^A y \\ 1 \end{pmatrix} = \begin{pmatrix} {}^A R_B & \mathbf{t} \\ \mathbf{0}_{1 \times 2} & 1 \end{pmatrix} \begin{pmatrix} {}^B x \\ {}^B y \\ 1 \end{pmatrix} \quad (2.10)$$

其中， $\mathbf{t} = (x, y)$ 代表坐标系的平移变换，而坐标系旋转变换用 ${}^A R_B$ 表示。因为 $\{A\}$ 和 $\{V\}$ 的轴是平行的，所以 ${}^A R_B = {}^V R_B$ 。将 \mathbf{P} 点的坐标向量用齐次形式表达为

$$\begin{aligned} {}^A \tilde{\mathbf{p}} &= \begin{pmatrix} {}^V R_B & \mathbf{t} \\ \mathbf{0}_{1 \times 2} & 1 \end{pmatrix} {}^B \tilde{\mathbf{p}} \\ &= {}^A T_B {}^B \tilde{\mathbf{p}} \end{aligned}$$

${}^A T_B$ 称为齐次转换矩阵。这个矩阵有一个非常特殊的结构，并且属于特殊的二维欧几里得群，即 $T \in SE(2) \subset \mathbb{R}^{3 \times 3}$ 。

① 参见附录D中有关向量、矩阵与线性代数的知识。

背景材料：一个向量 (x, y) 可以写成齐次形式 $\tilde{p} \in \mathbb{P}$, $\tilde{p} = (x_1, x_2, x_3)$, 其中 $x = x_1/x_3$, $y = x_2/x_3$, $x_3 \neq 0$ 。这里齐次向量的维度增加了一个, 在平面上的点就用三维向量表示。为了将点转换为齐次形式, 我们通常会附加一个 1, 变为 $\tilde{p} = (x, y, 1)$ 。字母上的波浪线表示该向量是齐次形式的。

齐次向量具有一个重要的属性, 即 \tilde{p} 等价于 $\lambda \tilde{p}$, $\forall \lambda \neq 0$, 记作 $\tilde{p} \simeq \lambda \tilde{p}$ 。这意味着 \tilde{p} 在平面中代表相同的点, 而与比例系数无关。在本书第四部分讨论计算机视觉时, 齐次表示法非常重要。更多细节见附录 I。

通过与方程(2.1)的比较, 很显然 ${}^A T_B$ 代表了相对位姿:

$$\xi(x, y, \theta) \sim \begin{pmatrix} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{pmatrix}$$

相对位姿 ξ 的一种具体表示是 $\xi \sim T \in SE(2)$, 以及 $T_1 \oplus T_2 \mapsto T_1 T_2$, 这是标准的矩阵乘法。

$$T_1 T_2 = \begin{pmatrix} R_1 & t_1 \\ 0_{1 \times 2} & 1 \end{pmatrix} \begin{pmatrix} R_2 & t_2 \\ 0_{1 \times 2} & 1 \end{pmatrix} = \begin{pmatrix} R_1 R_2 & t_1 + R_1 t_2 \\ 0_{1 \times 2} & 1 \end{pmatrix}$$

上文给出的代数运算规则之一是 $\xi \oplus 0 = \xi$ 。我们知道在矩阵中 $TI = T$, 其中 I 是单位矩阵, 所以对于位姿 0 对应的是 I 单位矩阵。另一条规则是 $\xi \ominus \xi = 0$ 。我们知道矩阵 $TT^{-1} = I$, 这意味着 $\ominus T \mapsto T^{-1}$

$$T^{-1} = \begin{pmatrix} R & t \\ 0_{1 \times 2} & 1 \end{pmatrix}^{-1} = \begin{pmatrix} R^T & -R^T t \\ 0_{1 \times 2} & 1 \end{pmatrix}$$

对于点 $\tilde{p} \in \mathbb{P}^2$, 有 $T \cdot \tilde{p} \mapsto T\tilde{p}$, 这是一个标准的矩阵-向量积。

为了使上述推导更形象具体, 下面我们将使用 MATLAB 工具箱展示一些具体数值化的例子。首先用函数 `se2` 创建一个齐次变换:

```
>> T1 = se2(1, 2, 30*pi/180)
T1 =
    0.8660    -0.5000    1.0000
    0.5000     0.8660    2.0000
         0         0    1.0000
```

它代表(1, 2)的平移和 30° 的旋转。相对于世界坐标系, 我们可以将该变换用绘图函数绘制如下:

```
>> axis([0 5 0 5]);
>> trplot2(T1, 'frame', '1', 'color', 'b')
```

绘图函数中的参数规定该坐标系的标签是 {1}, 且绘制成蓝色(图中 $Y_1 \{1\} X_1$ 坐标系), 如图 2.8 所示。我们再创建另一个平移(2, 1), 零旋转的相对位姿:

```
>> T2 = se2(2, 1, 0)
T2 =
     1     0     2
     0     1     1
     0     0     1
```

在图中用红色(图中 $Y_2 \{2\} X_2$ 坐标系)绘制出来:

```
>> hold on
>> trplot2(T2, 'frame', '2', 'color', 'r');
```

现在, 我们可以将两个相对位姿复合:

```
>> T3 = T1*T2
T3 =
    0.8660    -0.5000    2.2321
    0.5000     0.8660    3.8660
         0         0    1.0000
```


在图中用绿色(图中 $Y_3\{3\}X_3$ 坐标系)绘制出来:

```
>> trplot2(T3, 'frame', '3', 'color', 'g');
```

我们看到在复合中,坐标系 $\{2\}$ 的平移(2, 1)是相对于坐标系 $\{1\}$ 的。要注意的是,最终坐标系 $\{3\}$ 相对于世界坐标系的平移量并不是(3, 3),因为它是相对于旋转后的坐标系 $\{1\}$ 再平移的。复合运算的不可交换性可用以下例子证明:

```
>> T4 = T2*T1;
>> trplot2(T4, 'frame', '4', 'color', 'c');
```

可以看出,坐标系 $\{4\}$ 与坐标系 $\{3\}$ 不同。

现在我们相对于世界坐标系定义一个点(3, 2):

```
>> P = [3 ; 2];
```

这是一个列向量,把它加入图中:

```
>> plot_point(P, '*');
```

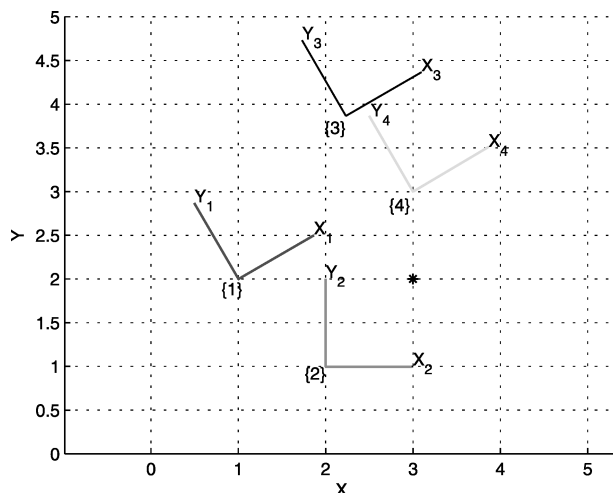


图 2.8 使用工具箱函数 trplot2 绘制出的坐标系

用式(2.1)确定该点相对于坐标系 $\{1\}$ 的坐标如下:

$${}^0\mathbf{p} = {}^0\xi_1 \cdot {}^1\mathbf{p}$$

将上式重新整理,得

$${}^1\mathbf{p} = {}^1\xi_0 \cdot {}^0\mathbf{p} = ({}^0\xi_1)^{-1} \cdot {}^0\mathbf{p}$$

下面代入具体数值:

```
>> P1 = inv(T1) * [P; 1]
P1 =
    1.7321
   -1.0000
    1.0000
```

首先,通过附加一个1将欧几里得点转换为齐次形式。该齐次形式的结果在坐标系 $\{1\}$ 中有一个负的y坐标。使用工具箱函数还可以进行反变换,操作如下:

```
>> h2e( inv(T1) * e2h(P) )
ans =
    1.7321
   -1.0000
```

这里又变回了欧几里得坐标点结果。辅助函数 `e2h` 将欧几里得坐标点转换为齐次形式，而 `h2e` 进行逆转换。更简洁的表达方式可写成

```
>> homtrans( inv(T1), P)
ans =
    1.7321
   -1.0000
```

该相同点相对于坐标系 {2} 有

```
>> P2 = homtrans( inv(T2), P)
P2 =
    1
    1
```

2.2 三维空间位姿描述

三维情况实际上是前一节讨论的二维情况的延伸。我们在二维坐标系上增加一个额外的坐标轴，通常用 z 表示，它同时与 x 轴和 y 轴正交。 z 轴的方向服从右手规则，并构成右手坐标系。与各坐标轴平行的单位向量记作 \hat{x} 、 \hat{y} 和 \hat{z} ①：

$$\hat{z} = \hat{x} \times \hat{y}, \quad \hat{x} = \hat{y} \times \hat{z}; \quad \hat{y} = \hat{z} \times \hat{x} \quad (2.11)$$

坐标系中的一个点 P 可用其 x 、 y 和 z 的坐标值 (x, y, z) 或者一个约束向量表示：

$$P = x\hat{x} + y\hat{y} + z\hat{z}$$

图 2.9 展示了一个相对于参考坐标系 {A} 的坐标系 {B}。可以清楚地看到，{B} 的原点通过向量 $t = (x, y, z)$ 进行平移，然后再通过某种复杂方式进行了旋转。正如二维情况一样，我们如何表示坐标系之间的方向是非常重要的。

我们的做法还是从相对于每个坐标系的任意一点 P 出发，然后再确定 ${}^A P$ 和 ${}^B P$ 之间的关系。我们仍然从两方面考虑：旋转和平移。三维情况下的旋转非常复杂，我们接下来在整个小节中都会对它进行讲解。

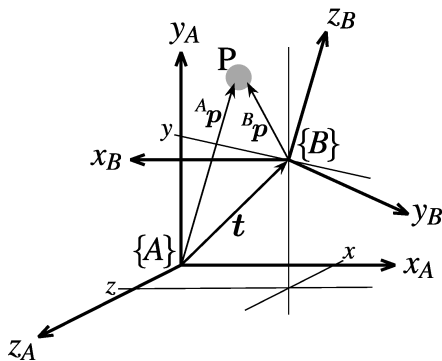


图 2.9 两个三维坐标系 {A} 和 {B}。
{B} 相对于 {A} 旋转和平移

2.2.1 三维空间姿态描述

任何两个独立的正交坐标系都可以通过一系列（不超过 3 次）相对于坐标轴的旋转联系起来，但其中连续的两次旋转不能绕同一轴线。

——欧拉旋转定理(奎伯斯, 1999)

图 2.9 中显示了两个右手坐标系，其方向差异很大，我们如何相对于一个坐标系去表示另一个坐标系呢？可以想象一下，将坐标系 {A} 拿在我们手里旋转，直到与坐标系 {B} 方向一致。欧拉旋转定理指出，任何旋转都可以看作是由一系列相对不同坐标轴的旋转组合而成的。

首先考虑绕单个坐标轴的旋转。图 2.10 显示了一个右手坐标系，以及它绕不同坐标轴旋转不同角度的情形。

① 在所有这些单位向量运算中，从左到右的符号排列（忽略等号）是以一种 xyz 的顺序进行循环置换的。

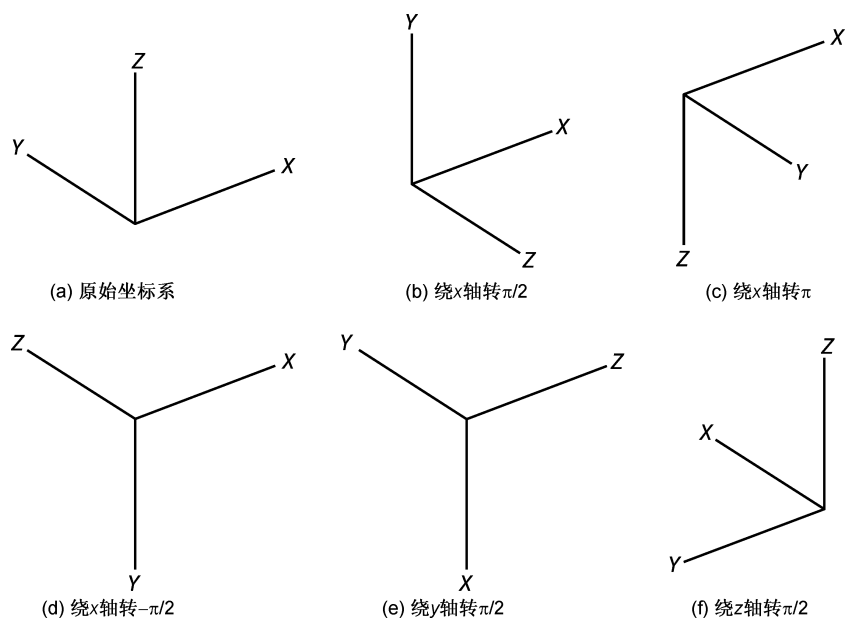


图 2.10 一个三维坐标系的旋转。(a)原始坐标系；(b)~(f)原始坐标系经各种旋转后的情况

但绕坐标轴连续旋转就没那么简单了，其中存在一些微妙的变化。如图 2.11 所示，它演示了一个坐标系按不同的顺序旋转的情形。可以看到，虽然旋转角度相同，但最终得到的坐标系方向完全不同，它取决于旋转的顺序。这是三维世界里一个深奥且让人迷惑的特征，而且它曾困扰了数学家们很长一段时间。我们在本章中使用位姿代数的意义就在于运算符 \oplus 是不可交换的，它体现了上述旋转顺序的重要性。

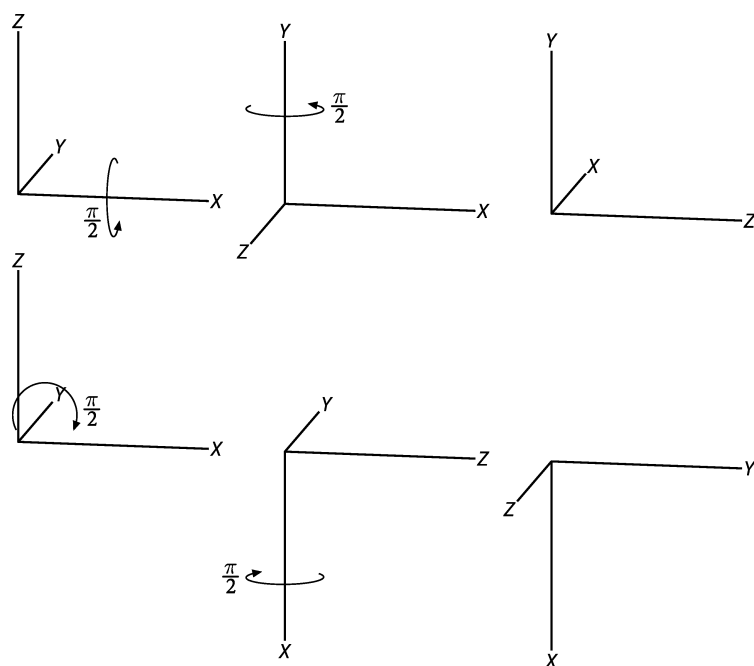
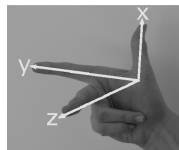


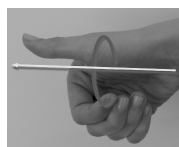
图 2.11 旋转顺序不可交换性的例子。在上面一排中坐标系先绕 x 轴旋转 $\pi/2$ ，然后绕 y 轴旋转 $\pi/2$ 。在下面一排旋转顺序正好颠倒了。两种旋转的结果明显不同

数学家已经提炼出许多表示旋转的方法，我们将在本节后续部分讨论其中几种：正交旋转矩阵，欧拉和卡尔丹角，旋转轴与角度，以及单位四元数。它们都可以表示为向量或矩阵，即 MATLAB 的自然数据类型，或者表示为一个工具箱自定义的类。工具箱中提供了许多函数可以将这些表示法相互转换，如图 2.15 所示。

背景材料：右手定则。所谓右手坐标系是指可以由右手前三个手指的指向确定，它们分别指向 x 、 y 和 z 轴的方向。



背景材料：绕一个向量的旋转。用你的右手抓握住该向量，并让拇指（你的 x 轴手指）指向向量箭头的方向，其余手指卷曲的指向就是角度增加的方向。



2.2.1.1 正交旋转矩阵

正如在二维情况下一样，我们可以用相对于参考坐标系的坐标轴单位向量表示它们所在坐标系的方向。每一个单位向量有 3 个元素，它们组成了 3×3 阶正交矩阵^A \mathbf{R}_B ：

$$\begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} = {}^A\mathbf{R}_B \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix} \quad (2.12)$$

上式将一个相对于坐标系 $\{B\}$ 的向量旋转为相对于坐标系 $\{A\}$ 的向量。矩阵 \mathbf{R} 属于特殊三维正交群，或 $\mathbf{R} \in SO(3) \subset \mathbb{R}^{3 \times 3}$ 。它具有前文提到的标准正交矩阵的特性，如 $\mathbf{R}^T = \mathbf{R}^{-1}$ 以及 $\det(\mathbf{R}) = 1$ 。

分别绕 x, y, z 轴旋转 θ 角后的标准正交旋转矩阵可表示为

$$\begin{aligned} \mathbf{R}_x(\theta) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \\ \mathbf{R}_y(\theta) &= \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \\ \mathbf{R}_z(\theta) &= \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{aligned}$$

工具箱中也提供了一些函数来计算这些基本的旋转矩阵，例如 $\mathbf{R}_x(\theta)$ 是

```
>> R = rotx(pi/2)
R =
    1.0000         0         0
         0    0.0000   -1.0000
         0    1.0000    0.0000
```

这个旋转的效果也显示在图 2.10(b)中。函数 `roty` 和 `rotz` 则分别用于计算 $\mathbf{R}_y(\theta)$ 和 $\mathbf{R}_z(\theta)$ 。

要绘制出相应的坐标系，可以用

```
>> trplot(R)
```

如图 2.12(a) 所示。如果要想使一个旋转更加生动有力，可以使用工具箱函数 `tranimate` 制作一个旋转动画：

```
>> tranimate(R)
```

它将展示世界坐标系旋转到指定坐标系的过程。

背景材料：在正交旋转矩阵中，从左至右的列向量可以告诉我们旋转后新坐标系各个轴相对于当前坐标系的方向。例如，如果

```
R =
    1.0000    0    0
    0    0.0000   -1.0000
    0    1.0000    0.0000
```

则新坐标系的 x 轴仍在以前 x 轴方向 $(1, 0, 0)$ ，但其 y 轴却在以前的 z 轴方向 $(0, 0, 1)$ ，而新的 z 轴在以前 y 轴的反方向 $(0, -1, 0)$ 。这里，之所以 x 轴没变，是因为这个旋转是围绕 x 轴发生的。行向量则相反，它们表示了当前坐标系的各个轴在新坐标系中的方向。

为了说明旋转的复合，我们继续旋转图 2.12(a) 所示的坐标系，这一次绕它的 y 轴旋转：

```
>> R = rotx(pi/2) * roty(pi/2)
R =
    0.0000    0    1.0000
    1.0000    0.0000   -0.0000
   -0.0000    1.0000    0.0000
>> trplot(R)
```

绘制出如图 2.12(b) 所示的坐标系。在这个坐标系中， x 轴现在指向世界坐标系 y 轴的方向。

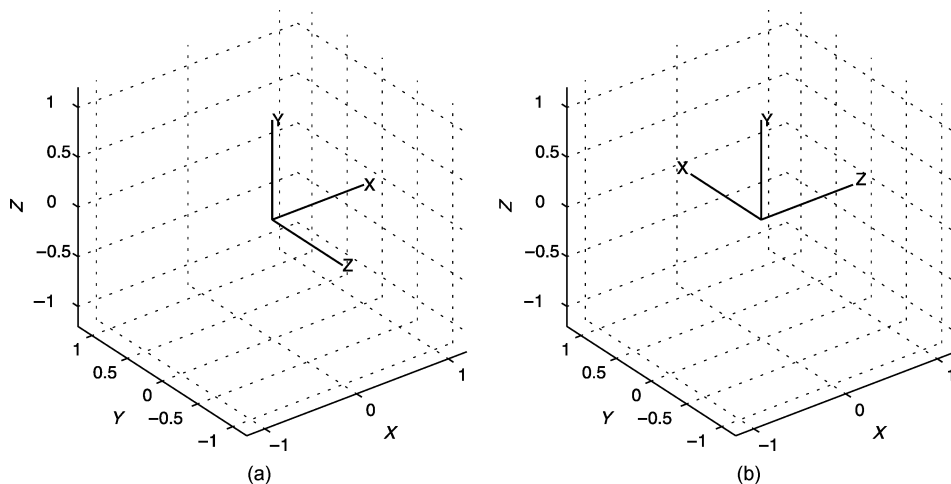


图 2.12 用函数 `trplot` 显示坐标系旋转。(a)参考坐标系绕 x 轴旋转 $\pi/2$ ；(b)坐标系 a 绕 y 轴旋转 $\pi/2$

如果颠倒以上旋转的顺序，可以看出旋转的不可交换性：

```
>> roty(pi/2)*rotx(pi/2)
ans =
    0.0000    1.0000    0.0000
    0    0.0000   -1.0000
   -1.0000    0.0000    0.0000
```

得到的结果是完全不同的。

回忆一下欧拉旋转定理，它指出任何旋转都可以用不超过 3 次绕坐标轴的旋转表示。这意味着，一般情况下两个坐标系之间的任意旋转均可分解为一组绕三个旋转轴转动的角度——这个问题将在下一节讨论。

正交矩阵有 9 个元素，但它们不是独立的。每一列都是单位长度，这提供了 3 个约束。列与列之间相互正交，又提供了另外 3 个约束^①。9 个元素加上 6 个约束，实际上只有 3 个独立的值。

2.2.1.2 三角度表示法

欧拉旋转定理要求绕 3 个轴依次旋转，但不能绕同一轴线连续旋转两次。旋转顺序分为两种：欧拉式和卡尔丹式，分别以欧拉和卡尔丹(Cardano)的名字命名。



背景材料：莱昂哈德·欧拉 (Leonhard Euler, 1707—1783)，瑞士著名数学家和物理学家，主导 18 世纪的数学研究。他是约翰·伯努利(Johan Bernoulli)的学生。他应用新的数学技术，如微积分，解决了力学和光学的许多问题。他还创造了我们现在使用的函数符号， $y = F(x)$ 。在机器人学方面，我们使用了他的旋转定理以及关于转动动力学的运动方程。

他非常多产，所写专著有 75 部，而且几乎一半的作品都是他在完全失明后生命的最后 17 年所著。

欧拉式是绕一个特定的轴重复旋转，但不是连续的：XYX、XZX、YXY、YZY、ZXZ 或 ZYZ。卡尔丹式的特点是绕 3 个不同轴旋转：XYZ、XZY、YZX、YXZ、ZXY 或 ZYX。一般来说，所有这些序列均被统称为欧拉角，共有 12 种形式可供选择。

现在通常是将所有三角度表示法都称作欧拉角，但这种称呼不够确切，因为一共有 12 种不同形式的欧拉角可供选择。在某一特定技术领域中，所说欧拉角往往指一种特定的旋转角顺序。

ZYZ 序列的欧拉角表示为

$$\mathbf{R} = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi) \quad (2.13)$$

它常用在航空和机械动力学中，并在本书工具箱中使用。欧拉角是一个三维向量 $\mathbf{F} = (\phi, \theta, \psi)$ 。

例如，要计算 $\mathbf{F} = (0.1, 0.2, 0.3)$ 的等价旋转矩阵，我们可以这样写：

```
>> R = rotz(0.1) * roty(0.2) * rotz(0.3);
```

或采用更简便的方式：

```
>> R = eul2r(0.1, 0.2, 0.3)
R =
    0.9021    -0.3836    0.1977
    0.3875     0.9216    0.0198
   -0.1898     0.0587    0.9801
```

上述问题的逆命题是要找到给定旋转矩阵的欧拉角，可采用以下函数：

```
>> gamma = tr2eul(R)
gamma =
    0.1000    0.2000    0.3000
```

^① 如果矩阵列向量是 \mathbf{c}_i ， $i \in 1 \cdots 3$ ，那么 $\mathbf{c}_1 \cdot \mathbf{c}_2 = \mathbf{c}_2 \cdot \mathbf{c}_3 = \mathbf{c}_3 \cdot \mathbf{c}_1 = 0$ 。

但是, 如果 θ 为负时, 如

```
>> R = eul2r(0.1, -0.2, 0.3)
R =
    0.9021    -0.3836    -0.1977
    0.3875     0.9216    -0.0198
    0.1898    -0.0587     0.9801
```

其反函数结果为

```
>> tr2eul(R)
ans =
   -3.0416     0.2000    -2.8416
```

它返回了一个正的 θ 值和两个不同的 ϕ 、 ψ 值。但这组欧拉角对应的旋转矩阵仍与前一组是相同的:

```
>> eul2r(ans)
ans =
    0.9021    -0.3836    -0.1977
    0.3875     0.9216    -0.0198
    0.1898    -0.0587     0.9801
```

两组不同的欧拉角对应同一个旋转矩阵, 说明从旋转矩阵到欧拉角的映射是不唯一的, 而工具箱函数返回的角度 θ 始终为正。

对于 $\theta=0$ 的情况:

```
>> R = eul2r(0.1, 0, 0.3)
R =
    0.9211    -0.3894         0
    0.3894     0.9211         0
         0         0     1.0000
```

反函数返回的角度值为

```
>> tr2eul(R)
ans =
         0         0     0.4000
```

它与原值完全不同。其实在这种情况下, 从式(2.13)得出的旋转矩阵是

$$\mathbf{R} = \mathbf{R}_z(\phi)\mathbf{R}_z(\psi) = \mathbf{R}_z(\phi + \psi)$$

由于 $\mathbf{R}_y = \mathbf{I}$, 所以可简化为只是 $\phi + \psi$ 的一个函数。对于逆运算而言, 只能确定这个和的值。要想得到其中每个值, 只能按惯例取 $\phi = 0$ 。 $\theta = 0$ 的情况实际上是一个奇异点, 我们将在下一节详细讨论。

另一种广泛使用的旋转角顺序是横滚-俯仰-偏航角, 即

$$\mathbf{R} = \mathbf{R}_x(\theta_r)\mathbf{R}_y(\theta_p)\mathbf{R}_z(\theta_y) \quad (2.14)$$

它用于描述船舶、飞机和车辆的姿态时非常直观。横滚、俯仰和偏航(也称为侧倾、姿态和航向)是指分别绕 x 、 y 、 z 轴的旋转。这个 xyz 角序列, 即专业上的卡尔丹角, 也被称为泰特-布莱恩角(Tait-Bryan)或导航角^①。对于航空及地面车辆而言, 通常定义 x 轴为前进的方向、 z 轴垂直向下、 y 轴指向右手方向^②。例如:

① 根据苏格兰物理学家、四元数支持者彼得·泰特(Peter Tait)和早期的威尔斯空气动力学家乔治·布莱恩(George Bryan)的名字命名。

② 注意, 大多数机器人学教科书[Paul 1981; Siciliano et al. 2008; Spong et al. 2006]都交换了其中的 x 轴和 z 轴, 把向前的方向定义为 z 轴方向而不是 x 轴方向, 也就是说绕 z 轴而不是 x 轴横滚。在第8版之前的机器人工具箱也采用这个顺序。但现在的工具箱中默认的是 xyz 顺序, 不过也可以用 'zyx' 选项来指定 zyx 的顺序。

```
>> R = rpy2r(0.1, 0.2, 0.3)
R =
    0.9363    -0.2896    0.1987
    0.3130    0.9447   -0.0978
   -0.1593    0.1538    0.9752
```

其逆运算为

```
>> gamma = tr2rpy(R)
gamma =
    0.1000    0.2000    0.3000
```

横滚-俯仰-偏航序列允许每个角度值有任意正负号,不会产生多解的情况。但它也有一个奇异点,即当 $\theta_p = \pm \pi/2$ 时,不过这个点刚好在大多数车辆可能的姿态范围以外。

背景材料: 吉罗拉莫·卡尔丹 (Girolamo Cardano, 1501—1576) 是意大利文艺复兴时期的数学家、医生、占星师,外加赌徒。他出生在意大利的帕维亚,是一个极有数学天赋的律师的私生子。他曾在帕多瓦大学学习医学,不久之后便成为诊断伤寒病的第一人。赌博也是他养活自己的一个方式,他写了一本关于机会博弈的书 *Liber de ludo aleae* (《论赌博游戏》),是有关概率问题的第一个系统论述,其中还包含了有效的作弊方法。他的家庭存在很多问题:他的长子因毒害其妻子受到刑罚,他的女儿是一名妓女,死于梅毒(他因此曾写过一篇关于梅毒治疗的论文)。他还计算并发布了耶稣的星座,但被人指责为异端并因此入狱,直到他发誓并放弃了他的教授职位才获释。



他在 1545 年出版的书 *ARS MAGNA* 中公布了三次和四次方程的解法,他还发明了密码锁,以及由三个同心圆组成的万向支架,可以让罗盘或陀螺仪自由转动(见图 2.13),他的发明还包括具有万向节的卡尔丹轴,这种轴至今仍在汽车上使用。

2.2.1.3 奇异点及万向节锁

上面叙述的三旋转角度表示方式中,一个根本的问题是奇异点。当中间的旋转轴平行于第一个或第三个旋转轴时这种情况就会发生。对于万向节锁(因电影《阿波罗 13 号》而出名的术语),也存在同样的问题。

用于导航的机械陀螺仪如图 2.13 所示。在其最核心的装配结构中有 3 个相互正交的框架,它们能使安装于其中的稳定体相对于宇宙静止。陀螺仪通过这个万向节机构连接到飞船机体上,这样无论飞船做任何机动飞行,都不会给陀螺仪内部的稳定平台施加外力矩。通过测量这些万向框架的轴相对于稳定平台的转动角度,就可以确定飞船的航行姿态——直接显示出飞船的横滚-俯仰-偏航角,图示设计中是卡尔丹角的 YZX 序列^①。

现在考虑陀螺仪中间万向架旋转角(相对于飞船的 z 轴旋转)为 90° 时的情况。这时陀螺仪的内万向架与外万向架的轴对齐,它们的旋转轴线重合。因为这两个旋转轴平行,这时陀螺仪只有两个有效的旋转轴,而不是原来的三个——我们称之为丢失了一个自由度^②。

① “登月舱坐标系是右手坐标系, +X 轴朝上, +Z 轴朝前, +Y 轴指向右边。旋转变换矩阵由一个基于 2-3-1 顺序的欧拉角构成,即:首先关于 Y 轴倾斜,然后绕 Z 轴滚动,最后关于 X 轴俯仰。正旋转分别是上倾,右滚,左偏” [Hoag 1963]。

② 该问题对于这个特殊陀螺仪的应用是一个明显的制约因素 [Hoag 1963]。这个问题可以通过增加第四个万向支架得以缓解,该方法在别的航天器上曾使用,但在阿波罗登月舱上因为重量和空间的原因并未采用。

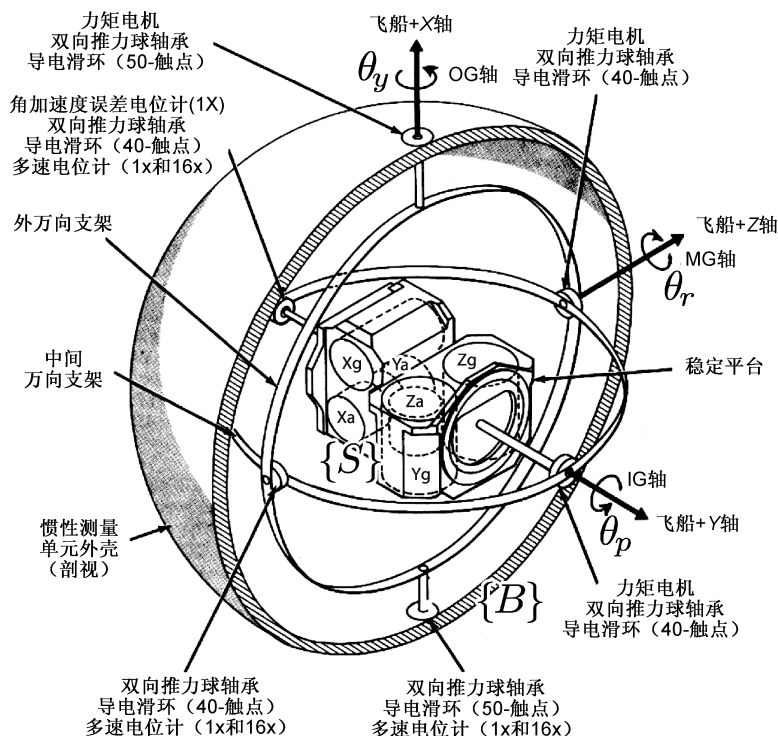


图 2.13 阿波罗月球舱 (LM) 使用的惯性测量单元 (IMU) 原理图。该飞船的坐标系定义为 x 轴向上, z 轴向前, y 轴向右。从稳定平台坐标系 $\{S\}$ 开始, 逐级向外到飞船体坐标系 $\{B\}$, 其旋转角的顺序是 YZX 。被标记为 X_g 、 Y_g 和 Z_g 的元素分别是绕 x 、 y 和 z 轴的角加速度传感器, 而被标记为 X_a 、 Y_a 和 Z_a 的元素分别是沿 x 、 y 和 z 轴的线性加速度传感器 (阿波罗操作手册, LMA790-3-LM)

从数学 (而非机械) 上看, 这个问题可以通过建立一种登月舱坐标系来解释, 其中固联在飞船机体上的坐标系 $\{B\}$ 相对于固联在稳定平台上的坐标系 $\{S\}$ 做旋转, 并且可以表示为

$${}^S R_B = R_y(\theta_p) R_z(\theta_r) R_x(\theta_y)$$

当 $\theta_r = \pi/2$ 时, 可以应用循环旋转规则得到下面的恒等式:^①

$$R_y(\theta) R_z(\frac{\pi}{2}) \equiv R_z(\frac{\pi}{2}) R_x(\theta)$$

进而得到

$${}^S R_B = R_z(\frac{\pi}{2}) R_x(\theta_p) R_x(\theta_y) = R_z(\frac{\pi}{2}) R_x(\theta_p + \theta_y)$$

上式中没有表示出飞船绕 y 轴的旋转。这就带来了问题, 因为航天器绕 y 轴的旋转将使稳定平台也旋转, 从而破坏其相对于恒星的精确对准。

① 旋转须服从循环旋转规则:

$$R_x(\pi/2) R_y(\theta) R_x(\pi/2)^T \equiv R_z(\theta)$$

$$R_y(\pi/2) R_z(\theta) R_y(\pi/2)^T \equiv R_x(\theta)$$

$$R_z(\pi/2) R_x(\theta) R_z(\pi/2)^T \equiv R_y(\theta)$$

以及反循环旋转规则:

$$R_y(\pi/2)^T R_x(\theta) R_y(\pi/2) \equiv R_z(\theta)$$

$$R_z(\pi/2)^T R_y(\theta) R_z(\pi/2) \equiv R_x(\theta)$$

一个自由度的缺失意味着在数学上我们不能反变换，我们只能建立两个角度之间的线性关系。在这种情况下，我们能做得最好的就是确定俯仰角和偏航角的总和。前面我们用欧拉角的奇异点也看到了类似的现象。

背景材料：任务时钟：02 08 12 47

- 飞行员：“去，盖德。”
- 盖德：“他越来越接近万向节锁了。”
- 飞行员：“收到。船长，建议他带上 C3, C4, B3, B4, C1 和 C2 的推进器，并通知他越来越接近万向节锁。”
- 船长：“收到。”

《阿波罗 13 号控制通信环任务》(1970)[Lovell and Kluger 1994, p131; NASA 1970]。



所有三角度形式的姿态表示，无论欧拉式或卡尔丹式，当两连续轴共线时都会遇到万向节锁同样的问题。对于 ZYZ 形式的欧拉角，它发生在 $\theta = k\pi$, $k \in \mathbb{Z}$ 时，对于用横滚-俯仰-偏航角的情况，会发生在 $\theta_p = \pm(2k+1)\pi/2$ 时。虽然都存在奇异点，但我们可以想办法让奇异点不在航行体正常运行时出现，这需要明智地选择角度序列和坐标系。

奇异点是采用最简化方法带来的一个不幸后果。为了消除这个问题，我们必须采取不同的姿态描述方法。其中，阿波罗登月舱团队的人提出一种用四个万向支架的系统，其成功的关键是引进了第四个参数，我们将很快在 2.2.1.6 节中讨论。

2.2.1.4 双向量表示法

对于关节臂式机器人，一般会在它的末端执行器上固联一个坐标系 $\{E\}$ ，如图 2.14 所示。通常情况下，工具的轴线为坐标系的 z 轴，并被称为接近向量，记为 $\hat{a} = (a_x, a_y, a_z)$ 。对于某些应用来说，定义接近向量比定义欧拉角或横滚-俯仰-偏航角更为方便。

然而定义出 z 轴的方向还不足以表示完整坐标系——我们还需要确定 x 轴和 y 轴的方向。为了确定末端执行器的姿态，我们定义一个与接近向量正交的向量，它位于机器人手爪的两个手指之间，被称为姿态向量，记为 $\hat{o} = (o_x, o_y, o_z)$ 。有了接近和姿态这两个单位向量，就足以完全确定旋转矩阵了：

$$R = \begin{pmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{pmatrix} \quad (2.15)$$

因为剩下的一列可以用式(2.11)计算，即 $\hat{n} = \hat{o} \times \hat{a}$ 。

即使两个向量 \hat{a} 和 \hat{o} 不是正交的，它们仍然能定义一个平面，而且计算出来的 \hat{n} 仍垂直于这个平面。在这种情况下，我们需要用 $\hat{o}' = \hat{a} \times \hat{n}$ 来计算一个新的向量值，它仍然在 \hat{a} 和 \hat{o} 定义的平面中，只不过与 \hat{a} 和 \hat{n} 都正交^①。

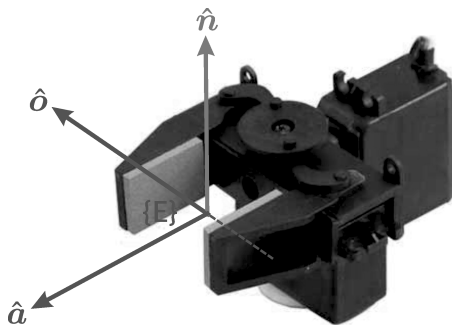


图 2.14 机器人末端执行器的坐标系用一个接近向量 \hat{a} 和一个姿态向量 \hat{o} 定义了它的位姿，从而可以计算出向量 \hat{n} 。 \hat{n} 、 \hat{o} 和 \hat{a} 分别对应于末端执行器坐标系的 x 轴、 y 轴和 z 轴（照片由 lynxmotion.com 提供）

① 只要它们是不平行的。