

第三节课习题

高翔

2017 年 12 月 18 日

1 习题说明

- 第 i 节课习题所有材料打包在 $L_i.zip$ 中, $\forall i = 1 \dots 8$ 。
- 习题分为若干种: **计算类**习题, 需要读者编程计算一个实际问题, 我们会附有参考答案以供自测。**操作类**习题, 会指导读者做一个具体的实验, 给出中间步骤截图或结果。**简述类**习题则提供阅读材料, 需要读者阅读材料后, 回答若干问题。
- 每个习题会有一些的分值。每次习题分值加和为 10 分。你需要获得 8 分以上才能得到“通过”的评价。带 * 的习题为附加题, 会在总分之外再提供一定的分值, 所以总和可能超过 10 分。换句话说, 你也可以选择一道附加题, 跳过一道正常题。
- 每道习题的给分由助教评判, 简述类习题可能存在一定开放性, 所以评分也存在主观因素。
- 请利用深蓝学院系统提交习题。每次习题我们会记通过与否。提交形式为 word 或 pdf 格式报告, 如有编程习题请提交可编译的源码。
- 为方便读者, 我通常会准备一些阅读材料, 放在 books/或 papers/目录下。请读者按个人需求使用这些材料。它们多数是从网络下载的, 如果侵犯到你的权利, 请及时告诉我。
- 每个习题会标注大致用时, 但视同学个人水平可能会有出入。
- 习题的完成情况会影响你对本课程内容的掌握程度, 请认真、独立完成。**习题总得分较高的同学将获得推荐资格。**

2 验证向量叉乘的李代数性质 (2 分, 约 1 小时)

我们说向量和叉乘运算构成了李代数, 现在请你验证它。书中对李代数的定义为: 李代数由一个集合 \mathbb{V} , 一个数域 \mathbb{F} 和一个二元运算 $[\cdot]$ 组成。如果它们满足以下几条性质, 称 $(\mathbb{V}, \mathbb{F}, [\cdot])$ 为一个李代数, 记作 \mathfrak{g} 。

1. 封闭性 $\forall \mathbf{X}, \mathbf{Y} \in \mathbb{V}, [\mathbf{X}, \mathbf{Y}] \in \mathbb{V}$.
2. 双线性 $\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{V}, a, b \in \mathbb{F}$, 有:

$$[a\mathbf{X} + b\mathbf{Y}, \mathbf{Z}] = a[\mathbf{X}, \mathbf{Z}] + b[\mathbf{Y}, \mathbf{Z}], \quad [\mathbf{Z}, a\mathbf{X} + b\mathbf{Y}] = a[\mathbf{Z}, \mathbf{X}] + b[\mathbf{Z}, \mathbf{Y}].$$

3. 自反性¹ $\forall \mathbf{X} \in \mathbb{V}, [\mathbf{X}, \mathbf{X}] = \mathbf{0}$.
4. 雅可比等价 $\forall \mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{V}, [\mathbf{X}, [\mathbf{Y}, \mathbf{Z}]] + [\mathbf{Y}, [\mathbf{Z}, \mathbf{X}]] + [\mathbf{Z}, [\mathbf{X}, \mathbf{Y}]] = \mathbf{0}$.

其中二元运算被称为**李括号**。

现取集合 $\mathbb{V} = \mathbb{R}^3$, 数域 $\mathbb{F} = \mathbb{R}$, 李括号为:

$$[\mathbf{a}, \mathbf{b}] = \mathbf{a} \times \mathbf{b}. \tag{1}$$

请验证 $\mathfrak{g} = (\mathbb{R}^3, \mathbb{R}, \times)$ 构成李代数。

¹ 自反性是指自己与自己的运算为零。

3 推导 SE(3) 的指数映射 (3 分, 约 1 小时)

课上给出了 SO(3) 的指数映射推导, 但对于 SE(3), 仅介绍了结论, 没有给出详细推导。请你完成 SE(3) 指数映射部分, 有关左雅可比详细推导。

设 $\xi = [\rho, \phi]^T \in \mathfrak{se}(3)$, 它的指数映射为:

$$\exp(\xi^\wedge) = \begin{bmatrix} \sum_{n=0}^{\infty} \frac{1}{n!} (\phi^\wedge)^n & \sum_{n=0}^{\infty} \frac{1}{(n+1)!} (\phi^\wedge)^n \rho \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (2)$$

令 $\rho = \theta \mathbf{a}$, 那么:

$$\sum_{n=0}^{\infty} \frac{1}{(n+1)!} (\phi^\wedge)^n = \frac{\sin \theta}{\theta} I + \left(1 - \frac{\sin \theta}{\theta}\right) \mathbf{a} \mathbf{a}^T + \frac{1 - \cos \theta}{\theta} \mathbf{a}^\wedge \triangleq \mathbf{J}. \quad (3)$$

这也正是课件里提到的左雅可比。

提示: 类比于 SO(3) 的泰勒展开, 然后合并奇偶数项级数即得。

4 伴随 (2 分, 约 1 小时)

在 $\text{SO}(3)$ 和 $\text{SE}(3)$ 上, 有一个东西称为伴随 (Adjoint)。下面请你证明 $\text{SO}(3)$ 伴随的性质。

对于 $\text{SO}(3)$, 有:

$$\mathbf{R} \exp(\mathbf{p}^\wedge) \mathbf{R}^\text{T} = \exp((\mathbf{R}\mathbf{p})^\wedge). \quad (4)$$

此时称 $\text{Ad}(\mathbf{R}) = \mathbf{R}$ 。

提示: 首先你需要证明 $\forall \mathbf{a} \in \mathbb{R}^3, \mathbf{R}\mathbf{a}^\wedge \mathbf{R}^\text{T} = (\mathbf{R}\mathbf{a})^\wedge$, 页面 <https://math.stackexchange.com/questions/2190603/derivation-of-adjoint-for-so3> 提示了一种简洁的途径。

对于 $\text{SE}(3)$, 有:

$$\mathbf{T} \exp(\boldsymbol{\xi}^\wedge) \mathbf{T}^{-1} = \exp((\text{Ad}(\mathbf{T})\boldsymbol{\xi})^\wedge) \quad (5)$$

其中 $\text{Ad}(\mathbf{T})$ 定义为:

$$\text{Ad}(\mathbf{T}) = \begin{bmatrix} \mathbf{R} & \mathbf{t}^\wedge \mathbf{R} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}. \quad (6)$$

这个性质将在后文的 Pose Graph 优化中用到。但是 $\text{SE}(3)$ 的证明较为复杂, 不作要求。

5 轨迹的描绘 (3 分, 约 1 小时)

我们通常会记录机器人的运动轨迹, 来观察它的运动是否符合预期。大部分数据集都会提供标准轨迹以供参考, 如 kitti、TUM-RGBD 等。这些文件会有各自的格式, 但首先你要理解它的内容。记世界坐标系为 W , 机器人坐标系为 C , 那么机器人的运动可以用 T_{WC} 或 T_{CW} 来描述。现在, 我们希望画出机器人在世界当中的运动轨迹, 请回答以下问题:

1. 事实上, T_{WC} 的平移部分即构成了机器人的轨迹。它的物理意义是什么? 为何画出 T_{WC} 的平移部分就得到了机器人的轨迹?
2. 我为你准备了一个轨迹文件 (code/trajectory.txt)。该文件的每一行由若干个数据组成, 格式为

$$[t, t_x, t_y, t_z, q_x, q_y, q_z, q_w],$$

其中 t 为时间, t_x, t_y, t_z 为 T_{WC} 的平移部分, q_x, q_y, q_z, q_w 是四元数表示的 T_{WC} 的旋转部分, q_w 为四元数实部。同时, 我为你提供了画图程序 draw_trajectory.cpp 文件。该文件提供了画图部分的代码, 请你完成数据读取部分的代码, 然后书写 CMakeLists.txt 以让此程序运行起来。注意我们需要用到 Pangolin 库来画图, 所以你需要事先安装 Pangolin (如果你做了第一次作业, 那么现在已经安装了)。CMakeLists.txt 可以参照 ORB-SLAM2 部分。

6 * 轨迹的误差 (2 分, 约 1 小时)

本题为附加题。

除了画出真实轨迹以外, 我们经常需要把 SLAM 估计的轨迹与真实轨迹相比较。下面说明比较的原理, 请你完成比较部分的代码实现。

设真实轨迹 (ground-truth) 为 \mathbf{T}_g , 估计轨迹 \mathbf{T}_e 。它们都以 \mathbf{T}_{WC} 的形式存储, 格式同上题。现在, 你需要计算估计轨迹的误差。我们假设每一个 \mathbf{T}_g 都与给定的 \mathbf{T}_e 对应。那么, 对于任意第 i 个位姿, 它的误差可定义为:

$$e_i = \|\log(\mathbf{T}_{gi}^{-1}\mathbf{T}_{ei})^\vee\|_2. \quad (7)$$

即两个位姿之差的李代数二范数。于是, 可以定义两条轨迹的均方根 (Root-Mean-Square-Error, RMSE) 误差为:

$$\text{RMSE}(g, e) = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2}. \quad (8)$$

我为你准备了 `code/ground-truth.txt` 和 `code/estimate.txt` 两条轨迹。请你根据上面公式, 实现 RMSE 的计算代码, 给出最后的 RMSE 结果。作为验算, 参考答案为: 2.207。

注:

1. 实际当中的轨迹比较还要更复杂一些。通常 ground-truth 由其他传感器记录 (如 vicon), 它的采样频率通常高于相机的步骤, 所以在处理之前还需要按照时间戳对齐。另外, 由于传感器坐标系不一致, 还需要计算两个坐标系之间的差异。这件事也可以用 ICP 解得, 我们将在后面的课程中讲到。
2. 你可以用上题的画图程序将两条轨迹画在同一个图里, 看看它们相差多少。