视觉SLAM理论与实践

第三次作业讲评

主讲人 田智豪

2 群的性质

1. {Z, +} 是否为群?若是,验证其满足群定义;若不是,说明理由。

2. {N, +} 是否为群?若是,验证其满足群定义;若不是,说明理由。
其中 Z 为整数集,N 为自然数集。

1. 是

2. 否, 逆不满足。

3 验证向量叉乘的李代数性质

深蓝学院
shenlanxueyuan.com

1. 封闭性
2. 双线性
3. 自反性

   上述三个通过差乘的基本性质就可以看出来。

4. 雅可比等价

   三矢量叉乘展开成点乘:

$$\forall X, Y, Z \in \mathbb{R}^3, X \times (Y \times Z) = (X \cdot Z) \cdot Y - (X \cdot Y) \cdot Z$$

参考：知乎搜“三矢量叉乘展开成点乘的公式如何证明？”

4 推导 SE(3) 的指数映射

已知性质:

1. 正弦和余弦函数的泰勒展开

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots \quad \text{for all } x$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots \quad \text{for all } x$$

2. SLAM14讲的(4.20)和(4.21)公式

$$a^\wedge a^\wedge = aa^T - I$$

$$a^\wedge a^\wedge a^\wedge = -a^\wedge$$

4 推导 SE(3) 的指数映射

推导:

$$\sum_{n=0}^{\infty} \frac{1}{(n+1)!}(\theta a^\wedge)^n = \frac{1}{1!}(\theta a^\wedge)^0 + \frac{1}{2!}(\theta a^\wedge)^1 + \frac{1}{3!}(\theta a^\wedge)^2 + \frac{1}{4!}(\theta a^\wedge)^3 + \frac{1}{5!}(\theta a^\wedge)^4 + \cdots$$

$$= I + \frac{\theta}{2!} a^\wedge + \frac{\theta^3}{4!}(a^\wedge)^3 + \frac{\theta^5}{6!}(a^\wedge)^5 + \cdots$$

$$+ \frac{\theta^2}{3!} a^\wedge a^\wedge + \frac{\theta^4}{5!}(a^\wedge)^4 + \frac{\theta^6}{7!}(a^\wedge)^6 + \cdots$$

$$= I + \frac{\theta}{2!} a^\wedge - \frac{\theta^3}{4!} a^\wedge + \frac{\theta^5}{6!} a^\wedge - \cdots$$

$$+ \frac{\theta^2}{3!} a^\wedge a^\wedge - \frac{\theta^4}{5!} a^\wedge a^\wedge + \frac{\theta^6}{7!} a^\wedge a^\wedge - \cdots$$

$$= I + \left(-\frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \cdots\right) \cdot \left(-\frac{a^\wedge}{\theta}\right)$$

$$+ \left(-\frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \cdots\right)\left(-\frac{a^\wedge a^\wedge}{\theta}\right)$$

4 推导 SE(3) 的指数映射

推导:

$$
\begin{aligned}
&= I + \left(1 - \frac{\theta^2}{2!} + \frac{\theta^4}{4!} - \frac{\theta^6}{6!} + \cdots\right)\cdot\left(-\frac{a^\wedge}{\theta}\right) - 1\cdot\left(-\frac{a^\wedge}{\theta}\right) \\
&\quad + \left(\theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \cdots\right)\cdot\left(-\frac{a^\wedge a^\wedge}{\theta}\right) - \theta\cdot\left(-\frac{a^\wedge a^\wedge}{\theta}\right) \\
&= I + \cos\theta\cdot\left(-\frac{a^\wedge}{\theta}\right) + \frac{a^\wedge}{\theta} \\
&\quad + \sin\theta\cdot\left(-\frac{a^\wedge a^\wedge}{\theta}\right) + a^\wedge a^\wedge \\
&= I + \frac{(1-\cos\theta)}{\theta} a^\wedge + \left(1 - \frac{\sin\theta}{\theta}\right)\left(aa^T - I\right) \\
&= I + \frac{1-\cos\theta}{\theta} a^\wedge + aa^T - I - \frac{\sin\theta}{\theta} aa^T + \frac{\sin\theta}{\theta} I \\
&= \frac{\sin\theta}{\theta} I + \left(1 - \frac{\sin\theta}{\theta}\right) aa^T + \frac{1-\cos\theta}{\theta} a^\wedge
\end{aligned}
$$

5 伴随

深蓝学院
shenlanxueyuan.com

按照题目中的提示一步步证明即可

1. 证明: $\forall \vec{a} \in \mathbb{R}^3,\ R\vec{a}^\wedge R^T = (Ra)^\wedge$

左边: $R\vec{a}^\wedge R^T R\vec{u} = R\vec{a}^\wedge \vec{u}$

$= R(\vec{a} \times \vec{u})$

右边 $(R\vec{a})^\wedge R\vec{u} = (R\vec{a}) \times (R\vec{u})$

$= R(\vec{a} \times \vec{u})$

2. 证明: $R\exp(p^\wedge)R^T = \exp((Rp)^\wedge)$

已知: $(Rp)^\wedge = Rp^\wedge R^T$

则: $\exp((Rp)^\wedge) = \exp(Rp^\wedge R^T)$

$= R\exp(p^\wedge)R^T$?

? 处使用性质: 如果T为可逆矩阵,

那么 $e^{TxT^{-1}} = Te^xT^{-1}$

证明如下: 已知 ① $T^{-1}(A+B)T = T^{-1}AT + T^{-1}BT$

② $(T^{-1}AT)^k = T^{-1}A^kT$

则 $Te^xT^{-1} = T\sum_{n=0}^{\infty}\frac{1}{n!}(x^\wedge)^n T^{-1}$

$= \sum_{n=0}^{\infty}\frac{1}{n!}T(x^\wedge)^n T^{-1}$

$= \sum_{n=0}^{\infty}\frac{1}{n!}(Tx^\wedge T^{-1})^n$

$= e^{TxT^{-1}}$

6 轨迹的描绘

6.1 事实上,Twc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 Twc 的平移部分就得到了机器人的轨迹?

Twc的物理意义: 机器人在某时刻相对于世界坐标系原点的位置。

因为在某个时间段内, 通过在每一个离散时刻机器人相对于世界坐标系的位置,我们可以估计出它的运动轨迹。轨迹的具体形式(折线或者曲线)取决于算法和离散时刻的间隔。

6 轨迹的描绘

深蓝学院
shenlanxueyuan.com

思路: 使用C++的标准库读入数据，并用Sophus库保存成SE3。

TODO部分的代码见右图。

```
string line;
double t,t_x,t_y,t_z,q_w,q_x,q_y,q_z;
ifstream myfile(trajectory_file);
if (!myfile)
{
cerr << "can't open the file" << endl;
exit(1);
}
while (!myfile.eof())
{
getline(myfile, line);
stringstream stream(line);
stream >> t >> t_x >> t_y >> t_z >> q_x >> q_y >> q_z >> q_w;
Eigen::Vector3d t(t_x, t_y, t_z);
Eigen::Quaterniond q(q_w,q_x,q_y,q_z);
q.normalize();

Sophus::SE3 SE3_qt(q,t);
poses.push_back(SE3_qt);
}
```

7 * 轨迹的误差

1. 使用Sophus按照公式计算RMSE

```
double RMSE(vector<Sophus::SE3, Eigen::aligned_allocator<Sophus::SE3>> T_g,
vector<Sophus::SE3, Eigen::aligned_allocator<Sophus::SE3>> T_i)
{
int T_g_length = T_g.size();
double errors = 0;
Eigen::Matrix<double,6,1> se3;

for(int i=0; i< T_g_length; i++)
{
se3 = (T_g[i].inverse()*T_i[i]).log();
errors += se3.norm()*se3.norm();
}
errors = sqrt(errors/T_g_length);
return errors;
}
```

7 * 轨迹的误差

2. 画出两个轨迹。修改画图函数,主要是使用了两个 for 循环对 Ti 和 Tg 的轨迹进行绘制:

```
for (size_t i = 0; i < T_g.size() - 1; i++) {
glColor3f(1 - (float) i / T_g.size(), 0.0f, (float) i / T_g.size());
glBegin(GL_LINES);
auto p1 = T_g[i], p2 = T_g[i + 1];
glVertex3d(p1.translation()[0], p1.translation()[1], p1.translation()[2]);
glVertex3d(p2.translation()[0], p2.translation()[1], p2.translation()[2]);
glEnd();
}

for (size_t i = 0; i < T_i.size() - 1; i++) {
glColor3f(1 - (float) i / T_i.size(), 0.0f, (float) i / T_i.size());
glBegin(GL_LINES);
auto p1 = T_i[i], p2 = T_i[i + 1];
glVertex3d(p1.translation()[0], p1.translation()[1], p1.translation()[2]);
glVertex3d(p2.translation()[0], p2.translation()[1], p2.translation()[2]);
glEnd();
}
```

感谢各位聆听！

Thanks for Listening