

[原代码](#)对新手似乎不是很友好，所以用Eigen重新实现了Fast-LIO的核心功能：[ghowoght/simple_lio](#)

本文中变量命名与Fast-LIO论文中不尽相同，比如为了便于阅读而忽略了很多上下标，如有误解之处，敬请指正。

0. 运算符定义

通过**指数/对数映射**可以实现李群和李代数之间的映射，定义 \boxplus 和 \boxminus 运算符，如下：

$$\begin{aligned}R \boxplus r &= R \exp(r) \\ R_1 \boxminus R_2 &= \text{Log}(R_2^T R_1) \\ a \boxplus b &= a + b \\ a \boxminus b &= a - b\end{aligned}$$

其中 $R, R_1, R_2 \in SO(3)$, $a, b \in \mathbb{R}^n$ 。指数映射为：

$$\text{Exp}(r) = I + \frac{r}{\|r\|} \sin(\|r\|) + \frac{r^2}{\|r\|^2} (1 - \cos(\|r\|))$$

上式也就是**罗德里格斯公式**，对数映射是它的逆映射。对应的程序如下：

```
// 指数映射
static Eigen::Matrix3d Exp(const Eigen::Vector3d& r){
    Eigen::Matrix3d expr;
    double theta = r.norm();
    if(theta < 1e-7){
        expr = Eigen::Matrix3d::Identity();
    }
    else{
        Eigen::Matrix3d skew = get_skew_symmetric(r / theta);
        expr = Eigen::Matrix3d::Identity() + sin(theta) * skew + (1 - cos(theta)) * skew * skew;
    }
    return expr;
}

// 对数映射
static Eigen::Vector3d Log(const Eigen::Matrix3d& R){
    double theta = (R.trace() > 3 - 1e-6) ? 0 : acos((R.trace() - 1) / 2);
    Eigen::Vector3d r(R(2,1) - R(1,2), R(0,2) - R(2,0), R(1,0) - R(0,1));
    return fabs(theta) < 0.001 ? (0.5 * r) : (0.5 * theta / sin(theta) * r);
}
```

1. 系统模型

因为fastlio论文看着有些晦涩，这里的推导主要参考高翔的博文[简明ESKF推导](#)

定义状态向量 x ：

$$x = [p^T \quad v^T \quad R^T \quad b_g^T \quad b_a^T \quad g^T]^T$$

状态的连续时间微分方程为：

$$\begin{aligned}
\dot{p} &= v \\
\dot{v} &= R(f^b - b_a - n_a) - g \\
\dot{R} &= R(\omega^b - b_g - n_g) \times \\
\dot{b}_g &= n_{bg} \\
\dot{b}_a &= n_{ba} \\
\dot{g} &= 0
\end{aligned}$$

其中 f^b 和 ω^b 分别为加速度计和陀螺仪测量值。

状态向量的估计值 \hat{x} ，表示为：

$$\hat{x} = \begin{bmatrix} \hat{p}^T & \hat{v}^T & \hat{R}^T & \hat{b}_g^T & \hat{b}_a^T & \hat{g}^T \end{bmatrix}^T$$

它的微分方程与真实值的微分方程相同，不过要忽略噪声。接下来推导误差状态向量 δx ，定义误差状态变量如下：

$$\begin{aligned}
p &= \hat{p} + \delta p \\
v &= \hat{v} + \delta v \\
R &= \hat{R} \delta R \\
b_g &= \hat{b}_g + \delta b_g \\
b_a &= \hat{b}_a + \delta b_a \\
g &= \hat{g} + \delta g
\end{aligned}$$

姿态部分的误差 δR 对应的李代数为 $\delta\theta$ ，即 $\delta R = Exp(\delta\theta)$ 。因此真实状态、估计状态、误差状态三者的关系可以表述为：

$$x = \hat{x} \oplus \delta x$$

$$\text{其中误差状态向量 } \delta x = [\delta p^T \quad \delta v^T \quad \delta\theta^T \quad \delta b_g^T \quad \delta b_a^T \quad \delta g^T]^T$$

误差状态中的**位置、零偏和重力项**都可以很容易的得到微分方程：

$$\begin{aligned}
\delta\dot{p} &= \delta v \\
\delta\dot{b}_g &= n_{bg} \\
\delta\dot{b}_a &= n_{ba} \\
\delta\dot{g} &= 0
\end{aligned}$$

速度、姿态均与 δR 相关，以下进行单独求导。

姿态误差

姿态真实值的微分方程：

$$\dot{R} = R(\omega^b - b_g - n_g) \times \quad (1)$$

姿态估计值的微分方程：

$$\dot{\hat{R}} = \hat{R}(\omega^b - \hat{b}_g) \times \quad (2)$$

姿态真实值、估计值、误差值三者的关系

$$R = \hat{R} Exp(\delta\theta) \quad (3)$$

对(3)式两侧分别求时间导数，得到：

$$\begin{aligned}
\dot{R} &= \dot{\hat{R}}Exp(\delta\theta) + \hat{R}\dot{Exp}(\delta\theta) \\
&= \dot{\hat{R}}Exp(\delta\theta) + \hat{R}Exp(\delta\theta)(\dot{\delta\theta}\times)
\end{aligned} \tag{4}$$

将(3)式代入(1)，得到：

$$\dot{R} = \hat{R}Exp(\delta\theta)(\omega^b - b_g - n_g) \times \tag{5}$$

联立(2)(4)(5)，得到：

$$\hat{R}(\omega^b - \hat{b}_g) \times Exp(\delta\theta) + \hat{R}Exp(\delta\theta)(\dot{\delta\theta}\times) = \hat{R}Exp(\delta\theta)(\omega^b - b_g - n_g) \times$$

消掉 \hat{R} ，再利用 $SO(3)$ 的伴随性质 $\phi \times R = R(R^T\phi)\times$ ，得到：

$$Exp(\delta\theta)(\dot{\delta\theta}\times) = Exp(\delta\theta)(\omega^b - b_g - n_g) \times -Exp(\delta\theta)(Exp(-\delta\theta)(\omega^b - \hat{b}_g)) \times$$

消掉 $Exp(\delta\theta)$ 以及 \times 符号，得到：

$$\begin{aligned}
\dot{\delta\theta} &= (\omega^b - b_g - n_g) - Exp(-\delta\theta)(\omega^b - \hat{b}_g) \\
&\approx (\omega^b - b_g - n_g) - (I - \delta\theta\times)(\omega^b - \hat{b}_g) \\
&= (\omega^b - (\hat{b}_g + \delta b_g) - n_g) - (I - \delta\theta\times)(\omega^b - \hat{b}_g) \\
&= -(\omega^b - b_g) \times \delta\theta - \delta b_g - n_g
\end{aligned}$$

速度误差

速度真实值的微分方程：

$$\dot{v} = R(f^b - b_a - n_a) - g \tag{1}$$

估计值的微分方程：

$$\dot{\hat{v}} = \hat{R}(f^b - \hat{b}_a) - \hat{g} \tag{2}$$

真实值、估计值、误差值的关系：

$$v = \hat{v} + \delta v \tag{3}$$

分别对(3)式两侧求导，得到：

$$\dot{v} = \dot{\hat{v}} + \dot{\delta v} \tag{4}$$

联立(1)(2)(4)，得到

$$\begin{aligned}
\hat{R}(f^b - \hat{b}_a) - \hat{g} + \dot{\delta v} &= R(f^b - b_a - n_a) - g \\
&= \hat{R}Exp(\delta\theta)(f^b - b_a - n_a) - g \\
&\approx \hat{R}(I + \delta\theta\times)(f^b - b_a - n_a) - g
\end{aligned}$$

化简得到：

$$\dot{\delta v} = \hat{R}(-\delta b_a - n_a - (f^b - b_a + \delta b_a - n_a) \times \delta\theta) - \delta g$$

忽略二次小量，得到：

$$\begin{aligned}
\dot{\delta v} &= -\hat{R}\delta b_a - \hat{R}(f^b - b_a) \times \delta\theta - \delta g - \hat{R}n_a \\
&= -\hat{R}\delta b_a - \hat{R}(f^b - b_a) \times \delta\theta - \delta g - n_a
\end{aligned}$$

状态方程

综上，误差状态的连续时间微分方程如下：

$$\begin{aligned}\delta\dot{p} &= \delta v \\ \delta\dot{v} &= -\hat{R}\delta b_a - \hat{R}(f^b - b_a) \times \delta\theta - \delta g - n_a \\ \delta\dot{\theta} &= -(\omega^b - b_g) \times \delta\theta - \delta b_g - n_g \\ \delta\dot{b}_g &= n_{bg} \\ \delta\dot{b}_a &= n_{ba} \\ \delta\dot{g} &= 0\end{aligned}$$

离散后的微分方程如下：

$$\begin{aligned}\delta p_{k+1} &= \delta p_k + \delta v \Delta t \\ \delta v_{k+1} &= v_k - \hat{R} \Delta t \delta b_a - \hat{R}(f^b - b_a) \Delta t \times \delta\theta - \delta g \Delta t - n_v \\ \delta\theta_{k+1} &= \text{Exp}(-(\omega^b - b_g) \Delta t) \delta\theta_k - \delta b_g \Delta t - n_\theta \\ \delta b_{g_{k+1}} &= \delta b_{g_k} + n_g \\ \delta b_{a_{k+1}} &= \delta b_{a_k} + n_a \\ \delta g_{k+1} &= g_k\end{aligned}$$

上式可以记为：

$$\delta x_{k+1} = F_x \delta x_k + F_w w$$

其中：

$$\begin{aligned}w &= [n_v^T \quad n_\theta^T \quad n_g^T \quad n_a^T]^T \sim N(0, Q) \\ F_x &= \begin{bmatrix} I & I\Delta t & 0 & 0 & 0 & 0 \\ 0 & I & -\hat{R}(f^b - b_a) \times & 0 & -\hat{R}\Delta t & I\Delta t \\ 0 & 0 & \text{Exp}(-(\omega^b - b_g) \Delta t) & -I\Delta t & 0 & 0 \\ 0 & 0 & 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 & 0 & I \end{bmatrix} \\ F_w &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ -I\Delta t & 0 & 0 & 0 \\ 0 & -I\Delta t & 0 & 0 \\ 0 & 0 & I\Delta t & 0 \\ 0 & 0 & 0 & I\Delta t \\ 0 & 0 & 0 & 0 \end{bmatrix}\end{aligned}$$

所以误差状态卡尔曼滤波的**预测部分**可以表述为：

$$\begin{aligned}\delta x_{k+1} &= F_x \delta x_k \\ P_{k+1} &= F_x P_k F_x^T + F_w Q F_w^T\end{aligned}$$

δx 在更新后都会被重置为0，因此可以不进行计算。

2. 观测模型

观测值

以平面特征点的观测方程为例，首先使用下式将lidar系下的平面特征点 p_l 转换到world系下，其中忽略了lidar和imu的外参以及运动补偿。

$$p_w = Rp_l + t$$

然后从地图中找到5个与 p_w 对应的平面特征点，用这些点拟合平面 $Ax + By + Cz + D = 0$ ，也即 $\frac{A}{D}x + \frac{B}{D}y + \frac{C}{D}z = -1$ 。求解得到平面的归一化法向量 $u = [A, B, C]^T$ 和截距 D 。然后计算点到平面的距离 $z = Ax + By + Cz + D$ 。

观测方程

如果使用真实的参数 x ，计算得到的点到平面的距离应该为0，也就是说：

$$\begin{aligned} h(x) &= h(\hat{x} \oplus \delta x) \\ &= u^T(Rp_l + t - q) \\ &= u^T((\hat{R} \oplus \delta R)p_l + \hat{t} + \tilde{t} - q) \\ &= 0 \end{aligned}$$

但是真实的参数无法获得，使用估计值 \hat{x} 计算出的距离为：

$$h(\hat{x}) = u^T(\hat{R}p_l + \hat{t} - q)$$

两者的偏差由姿态误差 δR 、平移误差 δt 和观测噪声引起。在 $\delta x = 0$ 处进行一阶泰勒展开

$$h(x) = h(\hat{x} \oplus \delta x) + v \approx h(\hat{x}) + H\delta x + v$$

H 是 $h(x)$ 在 $\delta x = 0$ 处的雅克比矩阵，变换上式得到：

$$H\delta x = h(\hat{x} \oplus \delta x) - h(\hat{x})$$

对等式两侧分别求偏导，得到

$$\begin{aligned} H &= \frac{\partial H\delta x}{\partial \delta x} \\ &= \frac{\partial(h(\hat{x} \oplus \delta x) - h(\hat{x}))}{\partial \delta x} \\ &= \lim_{\delta x \rightarrow 0} \frac{u^T(\hat{R}Exp(\delta\theta)p_l + \hat{t} + \delta t) - u^T(\hat{R}p_l + \hat{t})}{\delta x} \end{aligned}$$

分别求偏导 $\frac{\partial H\delta x}{\partial \delta\theta}$ 和 $\frac{\partial H\delta x}{\partial \delta t}$ ：

$$\begin{aligned} \frac{\partial H\delta x}{\partial \delta\theta} &= \lim_{\delta\theta \rightarrow 0} \frac{u^T(\hat{R}Exp(\delta\theta)p_l + \hat{t}) - u^T(\hat{R}p_l + \hat{t})}{\delta\theta} \\ &= \lim_{\delta\theta \rightarrow 0} \frac{u^T(\hat{R}Exp(\delta\theta) - \hat{R})p_l}{\delta\theta} \\ &\approx \lim_{\delta\theta \rightarrow 0} \frac{u^T(\hat{R}(I + \delta\theta \times) - \hat{R})p_l}{\delta\theta} \\ &= \lim_{\delta\theta \rightarrow 0} \frac{u^T \hat{R} \delta\theta \times p_l}{\delta\theta} \\ &= \lim_{\delta\theta \rightarrow 0} \frac{-u^T \hat{R} p_l \times \delta\theta}{\delta\theta} \\ &= -u^T \hat{R} (p_l \times) \end{aligned}$$

$$\frac{\partial H\delta x}{\partial \delta t} = u^T$$

因此观测方程的雅克比矩阵表示为：

$$h_i = \begin{bmatrix} u_i^T & 0 & -u_i^T \hat{R}(p_{li} \times) & 0 & 0 & 0 \end{bmatrix}$$

$$H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_M \end{bmatrix}$$

3. 迭代更新

迭代卡尔曼滤波中的状态更新过程可以看做优化问题。

后验分布

上一节中，有

$$\begin{aligned} h(x) &= h(\hat{x}) + H\delta x + v \\ &= 0 \end{aligned}$$

其中 $v \sim N(0, R)$ ，因此有

$$\begin{aligned} -v &= h(\hat{x}) + H\delta x \\ &\sim N(0, R) \end{aligned}$$

先验分布

真实的误差状态 δx 与第 k 次迭代得到的误差状态 δx_k 之间的关系为

$$\delta x = x \ominus \hat{x} = \hat{x}_k \oplus \delta x_k \ominus \hat{x}$$

其中 \hat{x}_k 是第 k 次迭代得到的状态向量，在 $\delta x_k = 0$ 处进行一阶泰勒展开，得到：

$$\delta x \approx \hat{x}_k \ominus \hat{x} + J_k \delta x_k$$

J_k 是 δx 在 $\delta x_k = 0$ 处的雅克比矩阵。可以轻易地得到，**除了姿态误差外，其余项均为单位矩阵**，以下求解姿态误差的雅克比矩阵 $\frac{\delta \theta}{\delta \theta_k}$ 。

令真实值为 R ，对应的李代数为 ϕ ；预测值为 \hat{R} ，对应的李代数为 $\hat{\phi}$ ；误差状态为 δR ，对应的李代数为 $\delta \theta$ ；第 k 次迭代的预测值为 \hat{R}_k ，对应的李代数为 $\hat{\phi}_k$ ；误差为 $\delta \theta_k$ 。满足：

$$\begin{aligned} \delta \theta &= R \ominus \hat{R} \\ &= \hat{R}_k \oplus \delta \theta_k \ominus \hat{R} \\ &= \text{Log}(\hat{R}^T \hat{R}_k \text{Exp}(\delta \theta_k)) \\ &= \text{Log}(\text{Exp}(-\hat{\phi}) \text{Exp}(\hat{\phi}_k) \text{Exp}(\delta \theta_k)) \end{aligned}$$

由右乘BCH近似得到：

$$\delta \theta \approx A^{-1}(\hat{\phi}_k \ominus \hat{\phi}) \delta \theta_k + \hat{\phi}_k \ominus \hat{\phi}$$

右乘BCH公式：

$$\begin{aligned} \text{Log}(\text{Exp}(\phi_1) \text{Exp}(\phi_2)) &\approx A^{-1}(\phi_1) \phi_2 + \phi_1, \phi_2 \text{ 为小量} \\ A(\phi) &= I + \frac{\sin(\|\phi\|)}{\|\phi\|} \frac{(\phi \times)^2}{\|\phi\|^2} - \frac{1 - \cos(\|\phi\|)}{\|\phi\|} \frac{\phi \times}{\|\phi\|} \end{aligned}$$

注：Fast-LIO中给出的A阵是**左乘**BCH近似雅克比矩阵，**右乘**雅克比矩阵**自变量取负数**或者**转置**即可得到**左乘**雅克比矩阵（所以论文里的A阵都进行了转置）

$$A_l(\phi) = A_r(-\phi)$$

$$\text{或者 } A_l(\phi) = A_r^T(\phi)$$

(14讲4.3.1)

因此 $\frac{\delta\theta}{\delta\theta_k} = A^{-1}(\hat{\phi}_k \boxminus \hat{\phi})$ 。令 $\begin{cases} J_k = \text{diag}\{I, I, A^{-1}(\hat{\phi}_k \boxminus \hat{\phi}), I, I, I\} \\ d = \hat{x}_k \boxminus \hat{x} \end{cases}$ ，则：

$$\begin{aligned} \delta x &= d + J_k \delta x_k \\ &\sim N(0, P) \end{aligned}$$

求解最大后验估计

由于 $\begin{cases} z + H\delta x_k \sim N(0, Q) \\ d + J_k \delta x_k \sim N(0, P) \end{cases}$ 均服从高斯分布，**最大后验估计**等价于最小化**贝叶斯风险(BMSE)**，即：

$$\min_{\delta x_k} \|d + J_k \delta x_k\|_{P^{-1}}^2 + \|z_k + H\delta x_k\|_{R^{-1}}^2$$

将目标函数 ϵ 表示如下：

$$\begin{aligned} \epsilon &= \frac{1}{2} \|d + J_k \delta x_k\|_{P^{-1}}^2 + \frac{1}{2} \|z_k + H\delta x_k\|_{R^{-1}}^2 \\ &= \frac{1}{2} (d + J_k \delta x_k)^T P^{-1} (d + J_k \delta x_k) + \frac{1}{2} (z_k + H\delta x_k)^T R^{-1} (z_k + H\delta x_k) \end{aligned}$$

求 δx_k 的偏导，得到：

$$\frac{\partial \epsilon}{\partial \delta x_k} = (J^T P^{-1} J + H^T R^{-1} H) \delta x_k + J^T P^{-1} d + H^T R^{-1} z$$

令 $\frac{\partial \epsilon}{\partial \delta x_k} = 0$ ，得到：

$$\delta x_k = (J^T P^{-1} J + H^T R^{-1} H)^{-1} (-J^T P^{-1} d - H^T R^{-1} z) \quad (1)$$

令 $Q = (J^T P^{-1} J + H^T R^{-1} H)^{-1}$ ，由**矩阵求逆定理**，即 $(A^{-1} + BD^{-1}C)^{-1} = A - AB(D + CAB)^{-1}CA$ ，得到：

$$Q = (I - (J^T P^{-1} J)^{-1} H^T (H (J^T P^{-1} J)^{-1} H^T + R)^{-1} H) (J^T P^{-1} J)^{-1}$$

令 $K = (J^T P^{-1} J)^{-1} H^T (H (J^T P^{-1} J)^{-1} H^T + R)^{-1}$ ，此即**卡尔曼增益**。 Q 可表示为：

$$Q = (I - KH) (J^T P^{-1} J)^{-1} \quad (2)$$

令 $U = (J^T P^{-1} J)^{-1}$ ，联立 $\begin{cases} K = UH^T (HUH^T + R)^{-1} \\ Q = (I - KH)U \end{cases}$ ，得到：

$$Q = KRH^{-T} \quad (3)$$

将 $\begin{cases} Q = (I - KH) (J^T P^{-1} J)^{-1} \\ Q = KRH^{-T} \end{cases}$ 带入(1)式，得到

$$\begin{aligned} \delta x_k &= (J^T P^{-1} J + H^T R^{-1} H)^{-1} (-J^T P^{-1} d - H^T R^{-1} z) \\ &= Q (-J^T P^{-1} d - H^T R^{-1} z) \\ &= (I - KH) (J^T P^{-1} J)^{-1} (-J^T P^{-1} d) + KRH^{-T} (-H^T R^{-1} z) \\ &= -Kz - (I - KH) J^{-1} d \end{aligned}$$

更新当前迭代次数的状态 \hat{x}_{k+1} ：

$$\hat{x}_{k+1} = \hat{x}_k \boxplus \delta x_k$$

所有迭代完成后，更新状态：

$$\begin{aligned}\bar{x} &= \hat{x}_{k+1} \\ \bar{P} &= (I - KH)P\end{aligned}$$

卡尔曼增益变形

Fast-LIO利用矩阵求逆定理推导了一种**新的卡尔曼增益计算形式**

$$K = (P^{-1} + H^T R^{-1} H)^{-1} H^T R^{-1}$$

该方法将**矩阵求逆**运算的维数限制为**状态的维数**，而不是**观测点云的数量**，减少求逆的计算耗时。以下是推导过程。

由**矩阵求逆定理**：

$$(HPH^T + R)^{-1} = R^{-1} - R^{-1}H(P^{-1} + H^T R^{-1}H)^{-1}H^T R^{-1}$$

将上式代入到原始卡尔曼增益计算公式，得到：

$$\begin{aligned}K &= PH^T \underbrace{(HPH^T + R)^{-1}}_{\text{矩阵求逆定理}} \\ &= PH^T (R^{-1} - R^{-1}H(P^{-1} + H^T R^{-1}H)^{-1}H^T R^{-1}) \\ &= (PH^T - \underbrace{PH^T R^{-1}H}_{P(P^{-1} + H^T R^{-1}H) - I}) (P^{-1} + H^T R^{-1}H)^{-1} H^T R^{-1} \\ &= (PH^T - PH^T + (P^{-1} + H^T R^{-1}H)^{-1} H^T) R^{-1} \\ &= (P^{-1} + H^T R^{-1}H)^{-1} H^T R^{-1}\end{aligned}$$

参考

[FAST-LIO: A Fast, Robust LiDAR-inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter](#)

[LINS: A Lidar-Inertial State Estimator for Robust and Efficient Navigation](#)

[IEKF-based Visual-Inertial Odometry using Direct Photometric Feedback](#)

[How to compute H](#)

[FAST-LIO2简明公式推导](#)