VINS-Mono数学公式推导

贺一家

November 13, 2017

Abstract

该笔记主要是对港科大沈邵颉老师组的开源代码VINS-Mono中涉及到的IMU公式(预积分factor，视觉factor，各类雅克比)进行推导和整理，方便以后学习，欢迎大家讨论交流。

1 预备知识

该部分只列举几个十分有用的性质，主要是和四元数的求导有关的，至于四元数更全面的性质，请参见参考文献。

1.1 四元数的简单性质

定义四元数$\mathbf{q}$由一个实部q_w和一个三维向量$\mathbf{q}_v = q_x i + q_y j + q_z k$组成

$$\mathbf{q} = \begin{bmatrix} q_w \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_w \\ q_x \\ q_y \\ q_w \end{bmatrix} \tag{1}$$

四元数的加法跟普通向量的加法一样，但是乘法就复杂多了，两个四元数相乘

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - \mathbf{p}_v^T \mathbf{q}_v \\ p_w \mathbf{q}_v + q_w \mathbf{p}_v + [\mathbf{p}_v]_\times \mathbf{q}_v \end{bmatrix} \tag{2}$$

两个四元数的这种乘法和旋转矩阵的连乘效果是一样的

$$\mathbf{q}_{AC} = \mathbf{q}_{AB} \otimes \mathbf{q}_{BC}, \qquad \mathbf{R}_{AC} = \mathbf{R}_{AB}\mathbf{R}_{BC} \tag{3}$$

上面的操作是四元数对四元数进行操作，但我们常常也会去旋转一个三维向量$\mathbf{x}$，将其旋转到另一个坐标系中进行表示，这用旋转矩阵很容易做到，直接矩阵相乘就可以$\mathbf{x}_A = \mathbf{R}_{AB}\mathbf{x}_B$, 用四元数的话，就稍微复杂一点，一个四元数怎么去旋转一个三维向量呢? 通常把这个三维向量看成一个纯四元数，即实部为0，然后在利用四元数乘法

$$\begin{bmatrix} 0 \\ \mathbf{x}_A \end{bmatrix} = \mathbf{q}_{AB} \otimes \begin{bmatrix} 0 \\ \mathbf{x}_B \end{bmatrix} \otimes \mathbf{q}^*_{AB} = [\mathbf{q}^*_{AB}]_R[\mathbf{q}_{AB}]_L \begin{bmatrix} 0 \\ \mathbf{x}_B \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{R}_{AB}\mathbf{x}_B \end{bmatrix} \tag{4}$$

符号$[\cdot]_R, [\cdot]_L$代表right- and left-quaternion-product matrices. (中文名叫啥我也不知道)

1.2 四元数求导

1.2.1 四元数对时间求导,参考sola的文档[1]

$$\begin{aligned}
\dot{\mathbf{q}} &= \lim_{\Delta t \to 0} \frac{\mathbf{q}(t+\Delta t) - \mathbf{q}(t)}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\mathbf{q} \otimes \Delta\mathbf{q} - \mathbf{q}}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\mathbf{q} \otimes (\begin{bmatrix} 1 \\ \Delta\phi/2 \end{bmatrix} - \begin{bmatrix} 1 \\ \mathbf{0} \end{bmatrix})}{\Delta t} \\
&= \lim_{\Delta t \to 0} \frac{\mathbf{q} \otimes \begin{bmatrix} 0 \\ \Delta\phi/2 \end{bmatrix}}{\Delta t} \\
&= \frac{1}{2}\mathbf{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega} \end{bmatrix}
\end{aligned} \tag{5}$$

对于上式得到的微分形式，还可以利用四元数的左右乘变换，把$\boldsymbol{\omega}$项从左边换到右边，即

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega})\mathbf{q} = \frac{1}{2}\mathbf{q} \otimes \boldsymbol{\omega} \tag{6}$$

1.2.2 四元数对四元数求导,参考了Yanbin Jia的课件[2]

在讲解求导前，先讲下叉乘以及反对称矩阵。对于一个三维向量$\mathbf{u} = (u_1, u_2, u_3)^3$，定义符号$[\cdot]_\times$表示其反对称矩阵，即

$$[\mathbf{u}]_\times = \begin{bmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{bmatrix} \tag{7}$$

两个向量叉乘有如下性质：$\mathbf{u}\times\mathbf{v}=[\mathbf{u}]_\times\mathbf{v}=-[\mathbf{v}]_\times\mathbf{u}$，由该性质容易得到两个向量叉乘对其中一个向量的导数

$$\begin{aligned}\frac{\partial(\mathbf{u}\times\mathbf{v})}{\partial\mathbf{v}}&=[\mathbf{u}]_\times\\ \frac{\partial(\mathbf{u}\times\mathbf{v})}{\partial\mathbf{u}}&=-\frac{\partial(\mathbf{v}\times\mathbf{u})}{\partial\mathbf{u}}\\ &=-[\mathbf{v}]_\times\end{aligned}$$

SLAM中涉及到各种对pose的求导，其中涉及到的一些四元数对四元数的求导，如下

$$\begin{aligned}\frac{\partial(\mathbf{p}\otimes\mathbf{q})}{\partial\mathbf{p}}&=\frac{\partial\begin{bmatrix}p_wq_w-\mathbf{p}_v^T\mathbf{q}_v\\ p_w\mathbf{q}_v+q_w\mathbf{p}_v+[\mathbf{p}_v]_\times\mathbf{q}_v\end{bmatrix}}{\partial\mathbf{p}}\\ &=\begin{bmatrix}q_w & -\mathbf{q}_v^T\\ \mathbf{q}_v & q_w\mathbf{I}_3-[\mathbf{q}_v]_\times\end{bmatrix}\end{aligned}\tag{8}$$

同理，容易得到

$$\frac{\partial(\mathbf{p}\otimes\mathbf{q})}{\partial\mathbf{q}}=\begin{bmatrix}p_w & -\mathbf{p}_v^T\\ \mathbf{p}_v & p_w\mathbf{I}_3+[\mathbf{p}_v]_\times\end{bmatrix}\tag{9}$$

这是四元数对四元数求导，得到的是4×4的矩阵，实际计算过程中，这个几乎很少用到。因为旋转实际上只有三个自由度，大家都是对四元数的增量$\delta\boldsymbol{\theta}$求导，而不是对四元数本身。

1.2.3 四元数和向量的乘积对四元数增量求导

在SLAM中，常遇到的是用四元数旋转一个三维点向量得到一个新向量。

$$\mathbf{x}_a=\mathbf{q}_{ab}\mathbf{x}_b=\mathbf{q}_{ab}\otimes\mathbf{x}_b\otimes\mathbf{q}_{ab}^*\tag{10}$$

这时候常常需要计算新向量对四元数增量$\delta\boldsymbol{\theta}$的导数。

$$\frac{\partial\mathbf{x}_a}{\partial\delta\boldsymbol{\theta}}=\lim_{\delta\boldsymbol{\theta}\to0}\frac{(\mathbf{q}_{ab}\otimes\begin{bmatrix}1\\ \frac{1}{2}\delta\boldsymbol{\theta}\end{bmatrix})\otimes\mathbf{x}_b\otimes(\mathbf{q}_{ab}\otimes\begin{bmatrix}1\\ \frac{1}{2}\delta\boldsymbol{\theta}\end{bmatrix})^*-\mathbf{q}_{ab}\otimes\mathbf{x}_b\otimes\mathbf{q}_{ab}^*}{\delta\boldsymbol{\theta}}\tag{11}$$

上面这个表达式看起来很复杂，代码里计算的时候也很复杂。在实际运算过程中，没必要用四元数来旋转向量，而是用其对应的矩阵形式替代。另外四元数微小增量和矩阵增量之间的对应关系为

$$\mathbf{q}\otimes\begin{bmatrix}1\\ \frac{1}{2}\mathbf{u}\phi\end{bmatrix}\to\mathbf{R}\exp([\mathbf{u}\phi]_\times)$$

上面这个式子其实就是IMU的旋转递推更新公式的两种不同形式：四元数形式和旋转矩阵形式，在后面的章节会介绍他们的来历，现在知道就行。开始求导：

$$
\begin{aligned}
\frac{\partial \mathbf{x}_a}{\partial \delta \boldsymbol{\theta}} &= \lim_{\delta\theta\to 0} \frac{\mathbf{q}_{ab} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta} \end{bmatrix} \mathbf{x}_b - \mathbf{q}_{ab}\mathbf{x}_b}{\delta\boldsymbol{\theta}} \\
&= \lim_{\delta\theta\to 0} \frac{\mathbf{R}_{ab}\exp([\delta\boldsymbol{\theta}]_\times)\mathbf{x}_b - \mathbf{R}_{ab}\mathbf{x}_b}{\delta\boldsymbol{\theta}} \\
&\approx \lim_{\delta\theta\to 0} \frac{\mathbf{R}_{ab}(\mathbf{I}+[\delta\boldsymbol{\theta}]_\times)\mathbf{x}_b - \mathbf{R}_{ab}\mathbf{x}_b}{\delta\boldsymbol{\theta}} \\
&= \lim_{\delta\theta\to 0} \frac{\mathbf{R}_{ab}[\delta\boldsymbol{\theta}]_\times\mathbf{x}_b}{\delta\boldsymbol{\theta}} \\
&= \lim_{\delta\theta\to 0} \frac{-\mathbf{R}_{ab}[\mathbf{x}_b]_\times\delta\boldsymbol{\theta}}{\delta\boldsymbol{\theta}} \\
&= -\mathbf{R}_{ab}[\mathbf{x}_b]_\times
\end{aligned} \tag{12}
$$

注意，考虑到公式的编辑篇幅，为了对一些求导公式进行简化，这里做一些简单的约定。 比如求导公式：

$$
\frac{\partial \mathbf{x}_a}{\partial \delta \boldsymbol{\theta}} = \lim_{\delta\theta\to 0} \frac{\mathbf{R}_{ab}\exp([\delta\boldsymbol{\theta}]_\times)\mathbf{x}_b - \mathbf{R}_{ab}\mathbf{x}_b}{\delta\boldsymbol{\theta}}
$$

后续的章节中，会直接简写为

$$
\frac{\partial \mathbf{x}_a}{\partial \delta \boldsymbol{\theta}} = \frac{\partial \mathbf{R}_{ab}\exp([\delta\boldsymbol{\theta}]_\times)\mathbf{x}_b}{\partial \delta\boldsymbol{\theta}}
$$

2 预积分

IMU测量值的真实值为$\boldsymbol{\omega}, \mathbf{a}$, 测量值为$\tilde{\boldsymbol{\omega}}, \tilde{\mathbf{a}}$

$$
\tilde{\boldsymbol{\omega}}^b = \boldsymbol{\omega}^b + \mathbf{b}^g + \mathbf{n}^g \tag{13}
$$

$$
\tilde{\mathbf{a}}^b = \mathbf{q}_{bw}(\mathbf{a}^w + \mathbf{g}^w) + \mathbf{b}^a + \mathbf{n}^a \tag{14}
$$

这里的上标g表示gyro，a表示acc，w表示在世界坐标系world，上标b表示imu机体坐标系body。注意，上面的公式用的是$+g^w$，svo预积分、msckf的很多论文都是$-g^w$。因此，要注意沈老师VINS里代码重力的方向问题。 由简单的位移速度加速度的关系，再联立之前的四元数对时间的导数有

$$
\begin{aligned}
\dot{\mathbf{p}}_{wb_t} &= \mathbf{v}_t^w \\
\dot{\mathbf{v}}_t^w &= \mathbf{a}_t^w \\
\dot{\mathbf{q}}_{wb_t} &= \mathbf{q}_{wb_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\boldsymbol{\omega}^{b_t} \end{bmatrix}
\end{aligned} \tag{15}
$$

在忽略bias的情况下，IMU的body坐标系从i时刻b_i到j时刻b_j的位姿和速度利用下式可以积分得到

$$\begin{aligned}
\mathbf{p}_{wb_j} &= \mathbf{p}_{wb_i} + \mathbf{v}_i^w \Delta t + \int\int_{t\in[i,j]} (\mathbf{q}_{wb_t}\mathbf{a}^{b_t} - \mathbf{g}^w)\delta t^2 \\
\mathbf{v}_j^w &= \mathbf{v}_i^w + \int_{t\in[i,j]} (\mathbf{q}_{wb_t}\mathbf{a}^{b_t} - \mathbf{g}^w)\delta t \\
\mathbf{q}_{wb_j} &= \int_{t\in[i,j]} \mathbf{q}_{wb_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\boldsymbol{\omega}^{b_t} \end{bmatrix} \delta t
\end{aligned} \tag{16}$$

从等式(16)中可以轻易看出，第i时刻imu坐标系b_i在世界坐标系中的旋转矩阵$\mathbf{q}_{wb_i}$需要事先知道，并且它还参与了位移速度等各项运算。因此这个矩阵十分重要，但恰恰这个旋转矩阵并不是那么容易计算准确。

一个很简单的公式转换是把$\mathbf{q}_{wb_t}$拆成$\mathbf{q}_{wb_i}\otimes\mathbf{q}_{b_ib_t}$,代入上式就可以简化得到预积分模型。

$$\begin{aligned}
\mathbf{p}_{wb_j} &= \mathbf{p}_{wb_i} + \mathbf{v}_i^w \Delta t - \frac{1}{2}\mathbf{g}^w\Delta t^2 + \mathbf{q}_{wb_i}\int\int_{t\in[i,j]} (\mathbf{q}_{b_ib_t}\mathbf{a}^{b_t})\delta t^2 \\
\mathbf{v}_j^w &= \mathbf{v}_i^w - \mathbf{g}^w\Delta t + \mathbf{q}_{wb_i}\int_{t\in[i,j]} (\mathbf{q}_{b_ib_t}\mathbf{a}^{b_t})\delta t \\
\mathbf{q}_{wb_j} &= \mathbf{q}_{wb_i}\int_{t\in[i,j]} \mathbf{q}_{b_ib_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\boldsymbol{\omega}^{b_t} \end{bmatrix} \delta t
\end{aligned} \tag{17}$$

在上式中，积分项主要是依赖i时刻的姿态，而不是相对于世界坐标系的姿态，而i时刻的姿态相对更容易得到，比如t=0时刻，旋转矩阵是单位矩阵。把积分项单独提出来，就是所谓的预积分了。

$$\begin{aligned}
\boldsymbol{\alpha}_{b_ib_j} &= \int\int_{t\in[i,j]} (\mathbf{q}_{b_ib_t}\mathbf{a}^{b_t})\delta t^2 \\
\boldsymbol{\beta}_{b_ib_j} &= \int_{t\in[i,j]} (\mathbf{q}_{b_ib_t}\mathbf{a}^{b_t})\delta t \\
\mathbf{q}_{b_ib_j} &= \int_{t\in[i,j]} \mathbf{q}_{b_ib_t} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\boldsymbol{\omega}^{b_t} \end{bmatrix} \delta t
\end{aligned} \tag{18}$$

因此，将17这个公式和预积分公式一起整理下，有

$$\begin{bmatrix} \mathbf{p}_{wb_j} \\ \mathbf{v}_j^w \\ \mathbf{q}_{wb_j} \\ \mathbf{b}_j^a \\ \mathbf{b}_j^g \end{bmatrix} = \begin{bmatrix} \mathbf{p}_{wb_i} + \mathbf{v}_i^w\Delta t - \frac{1}{2}\mathbf{g}^w\Delta t^2 + \mathbf{q}_{wb_i}\boldsymbol{\alpha}_{b_ib_j} \\ \mathbf{v}_i^w - \mathbf{g}^w\Delta t + \mathbf{q}_{wb_i}\boldsymbol{\beta}_{b_ib_j} \\ \mathbf{q}_{wb_i}\mathbf{q}_{b_ib_j} \\ \mathbf{b}_i^a \\ \mathbf{b}_i^g \end{bmatrix} \tag{19}$$

最后两行是加速度bias和角速度bias

2.1 预积分误差的定义

根据方程(19)可以得到两个时刻之间的的误差定义函数，误差函数是用imu的预积分项作为测量值和状态量之间差异：

$$\mathbf{e}_B(x_i,x_j)=\begin{bmatrix}\mathbf{r}_p\\ \mathbf{r}_q\\ \mathbf{r}_v\\ \mathbf{r}_{ba}\\ \mathbf{r}_{bg}\end{bmatrix}=\begin{bmatrix}\mathbf{q}_{b_iw}(\mathbf{p}_{wb_j}-\mathbf{p}_{wb_i}-\mathbf{v}_i^w\Delta t+\frac{1}{2}\mathbf{g}^w\Delta t^2)-\boldsymbol{\alpha}_{b_ib_j}\\ 2[\mathbf{q}_{b_jb_i}\otimes(\mathbf{q}_{b_iw}\otimes\mathbf{q}_{wb_j})]_{xyz}\\ \mathbf{q}_{b_iw}(\mathbf{v}_j^w-\mathbf{v}_i^w+\mathbf{g}^w\Delta t)-\boldsymbol{\beta}_{b_ib_j}\\ \mathbf{b}_j^a-\mathbf{b}_i^a\\ \mathbf{b}_j^g-\mathbf{b}_i^g\end{bmatrix}_{15\times 1} \tag{20}$$

上面误差中位移，速度，偏置都是直接相减得到。第三项是关于四元数的旋转误差，其中$[\cdot]_{xyz}$表示只取四元数的虚部(x,y,z)组成的三维向量。它的计算看起来不直观，下面仔细讲讲为什么这么定义四元数之间的旋转误差。

先看看旋转矩阵R的误差是一般是怎么计算和优化的。

$$\begin{aligned}\mathbf{R}_{err}&=\mathbf{R}_1^T\mathbf{R}_2\\ \log(\mathbf{R}_{err})&=[\mathbf{u}\phi]_\times\end{aligned} \tag{21}$$

旋转矩阵误差最后要通过log变换得到轴角误差的三维向量$err=\mathbf{u}\phi$。同理，用四元素时的误差计算和上述步骤一样

$$\begin{aligned}\mathbf{q}^{err}&=\mathbf{q}_1^*\mathbf{q}_2\\ \log(\mathbf{q}^{err})&=\begin{bmatrix}\log||\mathbf{q}^{err}||\\ \mathbf{u}\phi/2\end{bmatrix},\quad \mathbf{u}=\frac{\mathbf{q}_v^{err}}{||\mathbf{q}_v^{err}||},\quad \frac{1}{2}\phi=\arctan(||\mathbf{q}_v^{err}||,q_w)\end{aligned} \tag{22}$$

四元数的误差通过log变换以后得到的向量包含了轴角信息，并且还是4维的，第一维竟然和四元数的模长有关系，而后面三维$\mathbf{u}\phi/2$和旋转矩阵表示的误差只差一个2倍的关系，误差只和轴角这个三维的向量有关系这是很直观的。因此，在实际使用中，一般使用归一化四元数$\mathbf{q}$, 那么上式可以进行简化，第一维可以丢弃。

首先归一化四元数有如下性质：

$$\mathbf{q}=\begin{bmatrix}\cos\frac{\phi}{2}\\ \mathbf{u}\sin\frac{\phi}{2}\end{bmatrix}=\cos\frac{\phi}{2}+\mathbf{u}\sin\frac{\phi}{2}=e^{\mathbf{u}\frac{\phi}{2}},\quad ||\mathbf{q}||=1 \tag{23}$$

因此，对其取log()以后有

$$\log\mathbf{q}=\log(e^{\mathbf{u}\frac{\phi}{2}})=\begin{bmatrix}0\\ \mathbf{u}\frac{\phi}{2}\end{bmatrix}=0+\mathbf{u}\frac{\phi}{2} \tag{24}$$

当角度很小时，很明显有$\mathbf{u}\frac{\phi}{2} = \mathbf{u}\sin\frac{\phi}{2}$, 说明归一化四元数的log()等于该四元数的虚部。所以为了简化计算，一般计算归一化误差四元数$\mathbf{q}^{err}$的log()时，等价于直接取它的虚部，即

$$\log(\mathbf{q}^{err}) = \mathbf{q}_v^{err} = [\mathbf{q}^{err}]_{xyz} \tag{25}$$

所以四元数的误差公式为

$$err = 2\log(\mathbf{q}^{err}) = 2[\mathbf{q}^{err}]_{xyz} = \mathbf{u}\phi \tag{26}$$

这和用旋转矩阵去计算误差是等价的。

上面公式是从两个矩阵或者四元数得到误差(微小增量)。反过来，旋转矩阵乘以微小增量能得到新矩阵，四元数叉乘微小增量能得到新四元数，我们可以得到

$$\mathbf{R}' = \mathbf{R}\exp([\mathbf{u}\phi]_\times)$$

对应的有

$$\mathbf{q}' = \mathbf{q} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{u}\phi \end{bmatrix}$$

这两个公式说明了微小旋转矩阵和微小四元数增量之间的关系。

定义了误差的计算公式，需要最小二乘来优化状态变量，一般是计算Guass-Newton来递推优化变量:

$$\begin{aligned} \mathbf{A}\delta\mathbf{x} &= \mathbf{b} \\ \mathbf{J}^T\Sigma^{-1}\mathbf{J}\delta\mathbf{x} &= -\mathbf{J}^T\Sigma^{-1}\mathbf{r}(\hat{\mathbf{x}}) \\ \hat{\mathbf{x}} &= \hat{\mathbf{x}} \oplus \delta\mathbf{x} \end{aligned} \tag{27}$$

其中，$\mathbf{J} = \frac{\partial \mathbf{r}(\hat{\mathbf{x}})}{\partial \delta\mathbf{x}}$是误差对当前状态量的雅克比，符号$\oplus$表示状态量利用微小增量进行更新。

3 预积分的递推计算及其协方差

前面的推导过程，预积分量都是用的连续积分形式，而imu测量值是离散量，所以上面的积分形式有很多处理方式，比如一阶欧拉，mid-point，四阶龙格库塔等方法。在沈老师的代码中，imu的积分值使用mid-point方法计算的，也就是说两个相邻时刻k到k+1的姿态运算是用两个时刻的测量值$\mathbf{a}, \boldsymbol{\omega}$的平均值来计算的。结合之前预积分公

式(18)，可以写出基于mid-point方法的预积分公式离散形式。

$$
\begin{aligned}
\boldsymbol{\omega} &= \frac{1}{2}((\boldsymbol{\omega}^{b_k} - \mathbf{b}_k^g + \mathbf{n}_k^g) + (\boldsymbol{\omega}^{b_{k+1}} - \mathbf{b}_k^g + \mathbf{n}_{k+1}^g)) \\
\mathbf{q}_{b_i b_{k+1}} &= \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} \\
\mathbf{a} &= \frac{1}{2}(\mathbf{q}_{b_i b_k}(\mathbf{a}^{b_k} - \mathbf{b}_k^a + \mathbf{n}_k^a) + \mathbf{q}_{b_i b_{k+1}}(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a + \mathbf{n}_{k+1}^g)) \\
\boldsymbol{\alpha}_{b_i b_{k+1}} &= \boldsymbol{\alpha}_{b_i b_k} + \boldsymbol{\beta}_{b_i b_k}\delta t + \frac{1}{2}\mathbf{a}\delta t^2 \\
\boldsymbol{\beta}_{b_i b_{k+1}} &= \boldsymbol{\beta}_{b_i b_k} + \mathbf{a}\delta t \\
\mathbf{b}_{k+1}^a &= \mathbf{b}_k^a + \mathbf{n}_{\mathbf{b}_k^a} \\
\mathbf{b}_{k+1}^g &= \mathbf{b}_k^g + \mathbf{n}_{\mathbf{b}_k^g}
\end{aligned}
\tag{28}
$$

上述公式是在代码里是依顺序求解的，当然在代码里白噪声项$\mathbf{n}$是没有的，另外需要注意到的是上面公式中，在计算$\boldsymbol{\omega}, \mathbf{a}$时，是假设两个相邻时刻的bias是一样的。

将其代入误差公式$\mathbf{e}_B$就能计算误差量。在优化的过程中，还需要知道这个误差量的协方差矩阵。误差公式(20)中，误差的不确定度就是预积分测量值的噪声引起的，所以需要把一段时间内预积分测量值的协方差矩阵给计算出来。

在这里，先引入协方差的递推传递性质，直接回顾Forster的预积分论文，假设各变量的不确定度定义为$\boldsymbol{\eta}_{ik} = [\delta\boldsymbol{\phi}_{ik}, \delta\mathbf{v}_{ik}, \delta\mathbf{p}_{ik}]$，测量噪声为$\mathbf{n}_k = [\mathbf{n}_k^g, \mathbf{n}_k^a]$。如果能够找到相邻时刻不确定度的递推方程：

$$
\boldsymbol{\eta}_{ik} = \mathbf{A}_{k-1}\boldsymbol{\eta}_{ik-1} + \mathbf{B}_{k-1}\mathbf{n}_{k-1} \tag{29}
$$

上述的不确定度的传递由两部分组成：当前时刻的不确定性传递给下一时刻，当前时刻测量噪声带来的不确定性传递给下一时刻。有了它，协方差矩阵能通过递推计算得到：

$$
\boldsymbol{\Sigma}_{ik} = \mathbf{A}_{k-1}\boldsymbol{\Sigma}_{ik-1}\mathbf{A}_{k-1}^T + \mathbf{B}_{k-1}\boldsymbol{\Sigma}_{\mathbf{n}}\mathbf{B}_{k-1}^T \tag{30}
$$

其中，$\boldsymbol{\Sigma}_{\mathbf{n}}$是测量噪声的协方差矩阵，方差从i时刻开始进行递推$\boldsymbol{\Sigma}_{ii} = \mathbf{0}$，这是很直观的，因为初始化时刻只需计算第一次的测量噪声协方差。

同样的，在VINS框架下，我们只需要根据mid-point的状态量递推公式，推导出不确定度的递推公式就能够计算协方差了：

$$\begin{bmatrix} \delta\boldsymbol{\alpha}_{b_{k+1}b'_{k+1}} \\ \delta\boldsymbol{\theta}_{b_{k+1}b'_{k+1}} \\ \delta\boldsymbol{\beta}_{b_{k+1}b'_{k+1}} \\ \delta\mathbf{b}^a_{k+1} \\ \delta\mathbf{b}^g_{k+1} \end{bmatrix} = F \begin{bmatrix} \delta\boldsymbol{\alpha}_{b_k b'_k} \\ \delta\boldsymbol{\theta}_{b_k b'_k} \\ \delta\boldsymbol{\beta}_{b_k b'_k} \\ \delta\mathbf{b}^a_k \\ \delta\mathbf{b}^g_k \end{bmatrix} + G \begin{bmatrix} \mathbf{n}^a_k \\ \mathbf{n}^g_k \\ \mathbf{n}^a_{k+1} \\ \mathbf{n}^g_{k+1} \\ \mathbf{n}_{\mathbf{b}^a_k} \\ \mathbf{n}_{\mathbf{b}^g_k} \end{bmatrix} \tag{31}$$

F, G为两个时刻间的协方差矩阵，我们直接用代码里的公式在这里罗列一下，后续会进行推导。

$$F = \begin{bmatrix} \mathbf{I} & f_{12} & \mathbf{I}\delta t & -\frac{1}{4}(\mathbf{q}_{b_i b_k} + \mathbf{q}_{b_i b_{k+1}})\delta t^2 & f_{15} \\ \mathbf{0} & I - [\boldsymbol{\omega}]_\times & \mathbf{0} & \mathbf{0} & -\mathbf{I}\delta t \\ \mathbf{0} & f_{32} & \mathbf{I} & -\frac{1}{2}(\mathbf{q}_{b_i b_k} + \mathbf{q}_{b_i b_{k+1}})\delta t & f_{35} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \tag{32}$$

$$G = \begin{bmatrix} \frac{1}{4}\mathbf{q}_{b_i b_k}\delta t^2 & g_{12} & \frac{1}{4}\mathbf{q}_{b_i b_{k+1}}\delta t^2 & g_{14} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{1}{2}\delta t & \mathbf{0} & \frac{1}{2}\delta t & \mathbf{0} & \mathbf{0} \\ \frac{1}{2}\mathbf{q}_{b_i b_k}\delta t & g_{32} & \frac{1}{2}\mathbf{q}_{b_i b_{k+1}}\delta t & g_{34} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \delta t \end{bmatrix} \tag{33}$$

$$\begin{aligned} f_{12} &= \frac{\partial \boldsymbol{\alpha}_{b_i b_{k+1}}}{\partial \delta\boldsymbol{\theta}_{b_k b'_k}} = -\frac{1}{4}(\mathbf{R}_{b_i b_k}[\mathbf{a}^{b_k} - \mathbf{b}^a_k]_\times \delta t^2 + \mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_k} - \mathbf{b}^a_k)]_\times(\mathbf{I} - [\boldsymbol{\omega}]_\times \delta t)\delta t^2) \\ f_{32} &= \frac{\partial \boldsymbol{\beta}_{b_i b_{k+1}}}{\partial \delta\boldsymbol{\theta}_{b_k b'_k}} = -\frac{1}{2}(\mathbf{R}_{b_i b_k}[\mathbf{a}^{b_k} - \mathbf{b}^a_k]_\times \delta t + \mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_k} - \mathbf{b}^a_k)]_\times(\mathbf{I} - [\boldsymbol{\omega}]_\times \delta t)\delta t) \\ f_{15} &= \frac{\partial \boldsymbol{\alpha}_{b_i b_{k+1}}}{\partial \delta\mathbf{b}^g_k} = -\frac{1}{4}(\mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_k} - \mathbf{b}^a_k)]_\times \delta t^2)(-\delta t) \\ f_{35} &= \frac{\partial \boldsymbol{\beta}_{b_i b_{k+1}}}{\partial \delta\mathbf{b}^g_k} = -\frac{1}{2}(\mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_k} - \mathbf{b}^a_k)]_\times \delta t)(-\delta t) \\ g_{12} &= \frac{\partial \boldsymbol{\alpha}_{b_i b_{k+1}}}{\partial \mathbf{n}^g_k} = g_{14} = \frac{\partial \boldsymbol{\alpha}_{b_i b_{k+1}}}{\partial \mathbf{n}^g_{k+1}} = -\frac{1}{4}(\mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_k} - \mathbf{b}^a_k)]_\times \delta t^2)(\frac{1}{2}\delta t) \\ g_{32} &= \frac{\partial \boldsymbol{\beta}_{b_i b_{k+1}}}{\partial \mathbf{n}^g_k} = g_{34} = \frac{\partial \boldsymbol{\beta}_{b_i b_{k+1}}}{\partial \mathbf{n}^g_{k+1}} = -\frac{1}{2}(\mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_k} - \mathbf{b}^a_k)]_\times \delta t^2)(\frac{1}{2}\delta t) \end{aligned} \tag{34}$$

其中$\mathbf{0}, \mathbf{I}$都是3×3的矩阵。上面的雅克比矩阵有些公式还是比较复杂的，特别是跟旋转量有关部分的求导。接下来的章节将对雅克比进行推导。

另外，上述雅克比是两个相邻时刻k到k+1之间的。实际上我们可以把所有连续时刻的雅克比累乘得到k+1对i时刻的雅克比

$$\mathbf{J}_{k+1} = F\mathbf{J}_k \tag{35}$$

3.1 雅克比

从预积分的计算公式可以看出，位移预积分量$\boldsymbol{\alpha}$和速度预积分量$\boldsymbol{\beta}$的计算形式很类似，他们对各变量的雅克比应该是很相似的。因此，为了节约篇幅，下面只对速度预积分量的雅克比进行推导。

3.1.1 速度预积分量的雅克比

(1)对上一时刻速度预积分量的雅克比f_{33}

$$f_{33} = \frac{\partial \boldsymbol{\beta}_{b_i b_{k+1}}}{\partial \delta \boldsymbol{\beta}_{b_k b_k'}} = \mathbf{I}_{3\times 3} \tag{36}$$

(2)对角度预积分量的雅克比f_{32}

$$\begin{aligned}
\boldsymbol{\beta}_{b_i b_{k+1}} &= \boldsymbol{\beta}_{b_i b_k} + \mathbf{a}\delta t \\
&= \boldsymbol{\beta}_{b_i b_k} + \frac{1}{2}(\mathbf{q}_{b_i b_k}(\mathbf{a}^{b_k} - \mathbf{b}_k^a) + \mathbf{q}_{b_i b_{k+1}}(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a))\delta t \\
&= \boldsymbol{\beta}_{b_i b_k} + \frac{1}{2}(\mathbf{q}_{b_i b_k}(\mathbf{a}^{b_k} - \mathbf{b}_k^a) + \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a))\delta t \\
\frac{\partial \boldsymbol{\beta}_{b_i b_{k+1}}}{\partial \delta \boldsymbol{\theta}_{b_k b_k'}} &= \frac{\partial \mathbf{a}\delta t}{\partial \delta \boldsymbol{\theta}_{b_k b_k'}} \\
&= \frac{\partial \frac{1}{2}(\mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_k b_k'} \end{bmatrix} (\mathbf{a}^{b_k} - \mathbf{b}_k^a) + \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_k b_k'} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a))\delta t}{\partial \delta \boldsymbol{\theta}_{b_k b_k'}} \\
&= \frac{\partial \frac{1}{2}(\mathbf{R}_{b_i b_k} \exp([\delta\boldsymbol{\theta}_{b_k b_k'}]_\times)(\mathbf{a}^{b_k} - \mathbf{b}_k^a) + \mathbf{R}_{b_i b_k} \exp([\delta\boldsymbol{\theta}_{b_k b_k'}]_\times) \exp([\boldsymbol{\omega}\delta t]_\times)(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a))\delta t}{\partial \delta \boldsymbol{\theta}_{b_k b_k'}}
\end{aligned} \tag{37}$$

上面的导数分成两部分进行求导

$$\begin{aligned}
\frac{\partial \mathbf{R}_{b_i b_k} \exp([\delta\boldsymbol{\theta}_{b_k b_k'}]_\times)(\mathbf{a}^{b_k} - \mathbf{b}_k^a)\delta t}{\partial \delta \boldsymbol{\theta}_{b_k b_k'}} &= \frac{\partial \mathbf{R}_{b_i b_k}(\mathbf{I} + [\delta\boldsymbol{\theta}_{b_k b_k'}]_\times)(\mathbf{a}^{b_k} - \mathbf{b}_k^a)\delta t}{\partial \delta \boldsymbol{\theta}_{b_k b_k'}} \\
&= \frac{\partial - \mathbf{R}_{b_i b_k}[(\mathbf{a}^{b_k} - \mathbf{b}_k^a)\delta t]_\times \delta\boldsymbol{\theta}_{b_k b_k'}}{\partial \delta \boldsymbol{\theta}_{b_k b_k'}} \\
&= -\mathbf{R}_{b_i b_k}[(\mathbf{a}^{b_k} - \mathbf{b}_k^a)\delta t]_\times
\end{aligned} \tag{38}$$

对于第二部分的求导，需要知道旋转矩阵乘以一个向量的反对称矩阵有如下性质：

$$[\mathbf{R}\mathbf{a}]_\times = \mathbf{R}[\mathbf{a}]_\times \mathbf{R}^T \tag{39}$$

第二部分结果为

$$\begin{aligned}
\frac{\partial \mathbf{R}_{b_i b_k} \exp([\delta\boldsymbol{\theta}_{b_k b'_k}]_\times) \exp([\boldsymbol{\omega}\delta t]_\times)(\mathbf{a}^{b_{k+1}} - \mathbf{b}^a_k)\delta t}{\partial \delta\boldsymbol{\theta}_{b_k b'_k}} &= \frac{\partial \mathbf{R}_{b_i b_k}(\mathbf{I} + [\delta\boldsymbol{\theta}_{b_k b'_k}]_\times) \exp([\boldsymbol{\omega}\delta t]_\times)(\mathbf{a}^{b_{k+1}} - \mathbf{b}^a_k))\delta t}{\partial \delta\boldsymbol{\theta}_{b_k b'_k}} \\
&= -\frac{\partial \mathbf{R}_{b_i b_k}[\exp([\boldsymbol{\omega}\delta t]_\times)(\mathbf{a}^{b_{k+1}} - \mathbf{b}^a_k)\delta t]_\times \delta\boldsymbol{\theta}_{b_k b'_k}}{\partial \delta\boldsymbol{\theta}_{b_k b'_k}} \\
&= -\frac{\partial \mathbf{R}_{b_i b_k} \exp([\boldsymbol{\omega}\delta t]_\times)[(\mathbf{a}^{b_{k+1}} - \mathbf{b}^a_k)\delta t]_\times \exp([-\boldsymbol{\omega}\delta t]_\times)\delta\boldsymbol{\theta}_{b_k b'_k}}{\partial \delta\boldsymbol{\theta}_{b_k b'_k}} \\
&= -\mathbf{R}_{b_i \color{red}{b_{k+1}}}[(\mathbf{a}^{b_k} - \mathbf{b}^a_k)\delta t]_\times \exp([-\boldsymbol{\omega}\delta t]_\times) \\
&\approx -\mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_k} - \mathbf{b}^a_k)\delta t]_\times (\mathbf{I} - [\boldsymbol{\omega}\delta t]_\times)
\end{aligned} \tag{40}$$

将上面两部分综合起来就能得到

$$f_{32} = \frac{\partial \boldsymbol{\beta}_{b_i b_{k+1}}}{\partial \delta\boldsymbol{\theta}_{b_k b'_k}} = -\frac{1}{2}(\mathbf{R}_{b_i b_k}[\mathbf{a}^{b_k} - \mathbf{b}^a_k]_\times \delta t + \mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_k} - \mathbf{b}^a_k)]_\times (\mathbf{I} - [\boldsymbol{\omega}]_\times \delta t)\delta t) \tag{41}$$

(3)对k时刻角速度bias的雅克比f_{35}

$$\begin{aligned}
\boldsymbol{\omega} &= \frac{1}{2}((\boldsymbol{\omega}^{b_k} - \mathbf{b}^g_k) + (\boldsymbol{\omega}^{b_{k+1}} - \mathbf{b}^g_k)) = \frac{1}{2}(\boldsymbol{\omega}^{b_k} + \boldsymbol{\omega}^{b_{k+1}}) - \mathbf{b}^g_k \\
\boldsymbol{\beta}_{b_i b_{k+1}} &= \boldsymbol{\beta}_{b_i b_k} + \frac{1}{2}(\mathbf{q}_{b_i b_k}(\mathbf{a}^{b_k} - \mathbf{b}^a_k) + \color{red}{\mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} (\mathbf{a}^{b_{k+1}} - \mathbf{b}^a_k))\delta t}
\end{aligned}$$

只有红色公式部分和角速度bias有关系，所以后面的雅克比计算只考虑红色公式部分。

$$
\begin{aligned}
f_{35} = \frac{\partial \boldsymbol{\beta}_{b_i b_{k+1}}}{\partial \delta \mathbf{b}_k^g} &= \frac{\partial \frac{1}{2}\mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(\boldsymbol{\omega} - \delta \mathbf{b}_k^g)\delta t \end{bmatrix} (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)\delta t}{\partial \delta \mathbf{b}_k^g} \\
&= \frac{1}{2} \frac{\partial \mathbf{R}_{b_i b_k} \exp\left([(\boldsymbol{\omega} - \delta \mathbf{b}_k^g)\delta t]_\times\right)(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)\delta t}{\partial \delta \mathbf{b}_k^g} \\
&= \frac{1}{2} \frac{\partial \mathbf{R}_{b_i b_k} \exp\left([-\delta \mathbf{b}_k^g \delta t]_\times\right) \exp\left([\mathbf{J}_r(-\delta \mathbf{b}_k^g \delta t)\boldsymbol{\omega}\delta t]_\times\right)(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)\delta t}{\partial \delta \mathbf{b}_k^g} \\
&\approx \frac{1}{2} \frac{\partial \mathbf{R}_{b_i b_k} \exp\left([-\delta \mathbf{b}_k^g \delta t]_\times\right) \exp\left([\boldsymbol{\omega}\delta t)]_\times (\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)\delta t}{\partial \delta \mathbf{b}_k^g} \\
&= \frac{1}{2} \frac{\partial \mathbf{R}_{b_i b_k} (\mathbf{I} + [\delta \mathbf{b}_k^g]_\times(-\delta t)) \exp\left([\boldsymbol{\omega}\delta t]_\times\right)(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)\delta t}{\partial \delta \mathbf{b}_k^g} \\
&= \frac{1}{2} \frac{\partial - \mathbf{R}_{b_i b_k} ([\exp\left([\boldsymbol{\omega}\delta t]_\times\right)(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)]_\times)\delta t(-\delta t)\delta \mathbf{b}_k^g}{\partial \delta \mathbf{b}_k^g} \\
&= \frac{1}{2} \frac{\partial - \mathbf{R}_{b_i b_k} \exp\left([\boldsymbol{\omega}\delta t]_\times\right)([(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)]_\times) \exp\left([-\boldsymbol{\omega}\delta t]_\times\right)\delta t(-\delta t)\delta \mathbf{b}_k^g}{\partial \delta \mathbf{b}_k^g} \\
&= -\frac{1}{2}\mathbf{R}_{b_i {\color{red} b_{k+1}}}([(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)]_\times) \exp\left([-\boldsymbol{\omega}\delta t]_\times\right)\delta t(-\delta t) \\
&= -\frac{1}{2}\mathbf{R}_{b_i b_{k+1}}([(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)]_\times)(\mathbf{I} - [\boldsymbol{\omega}\delta t]_\times)\delta t(-\delta t)
\end{aligned}
\tag{42}
$$

注意，代码里的公式为$-\frac{1}{2}(\mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_k} - \mathbf{b}_k^a)]_\times \delta t)(-\delta t)$，我们的推导结果和他的代码稍微有点不一样，估计他们把$\exp([-\boldsymbol{\omega}\delta t]_\times)$也做了单位矩阵的近似。

3.1.2 旋转预积分量的雅克比

旋转部分的公式为:

$$
\begin{aligned}
\boldsymbol{\omega} &= \frac{1}{2}((\boldsymbol{\omega}^{b_k} - \mathbf{b}_k^g) + (\boldsymbol{\omega}^{b_{k+1}} - \mathbf{b}_k^g)) \\
\mathbf{q}_{b_i b_{k+1}} &= \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} \\
&= \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(\frac{1}{2}(\boldsymbol{\omega}^{b_k} + \boldsymbol{\omega}^{b_{k+1}}) - \mathbf{b}_k^g)\delta t \end{bmatrix}
\end{aligned}
\tag{43}
$$

接下来先推导上一时刻的旋转误差$\delta\boldsymbol{\theta}_{b_k b'_k}$会怎么影响当前时刻的旋转误差$\delta\boldsymbol{\theta}_{b_{k+1} b'_{k+1}}$，假设两个时刻的真值为$\mathbf{q}_{b_i b_{k+1}}, \mathbf{q}_{b_i b_k}$，两个时刻间的增量真值为$\mathbf{q}_{b_k b_{k+1}}$。当只考虑旋转误差$\delta\boldsymbol{\theta}_{b_k b'_k}$的影响，而不考虑测量值角速度偏置

误差影响的时候，可以假设 $\mathbf{q}_{b_k b_{k+1}} \approx \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix}$。另外，三元组四元数相乘有如下性质，参考[3]公式(80)：

$$\mathbf{q} \otimes \mathbf{p} \otimes \mathbf{q}^* = [\mathbf{q}]_L [\mathbf{q}]_R^T \mathbf{p} = \begin{bmatrix} p_w \\ \mathbf{R}\mathbf{p}_v \end{bmatrix} \tag{44}$$

其中$\mathbf{R}$是和$\mathbf{q}$对应的旋转矩阵，现在开始正式推导：

$$\begin{aligned} \mathbf{q}_{b_i b_{k+1}} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_{k+1} b'_{k+1}} \end{bmatrix} &= \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_k b'_k} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} \\ \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_{k+1} b'_{k+1}} \end{bmatrix} &= \mathbf{q}^*_{b_i b_{k+1}} \otimes \mathbf{q}_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_k b'_k} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} \\ &= \mathbf{q}_{b_{k+1} b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_k b'_k} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} \\ &\approx \begin{bmatrix} 1 \\ -\frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_k b'_k} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{R}\delta\boldsymbol{\theta}_{b_k b'_k} \end{bmatrix} \end{aligned} \tag{45}$$

所以有

$$\begin{aligned} \delta\boldsymbol{\theta}_{b_{k+1} b'_{k+1}} &= \mathbf{R}\delta\boldsymbol{\theta}_{b_k b'_k} \\ &= \exp([-\boldsymbol{\omega}\delta t]_\times)\delta\boldsymbol{\theta}_{b_k b'_k} \\ &\approx (\mathbf{I} - [\boldsymbol{\omega}\delta t]_\times)\delta\boldsymbol{\theta}_{b_k b'_k} \\ \frac{\partial \delta\boldsymbol{\theta}_{b_{k+1} b'_{k+1}}}{\partial \delta\boldsymbol{\theta}_{b_k b'_k}} &= \mathbf{I} - [\boldsymbol{\omega}\delta t]_\times \end{aligned} \tag{46}$$

接下来，推导角速度bias误差的$\delta\mathbf{b}^g_k$影响，首先回顾公式

$$\mathbf{p} \otimes \mathbf{q} = \begin{bmatrix} p_w q_w - \mathbf{p}_v^T \mathbf{q}_v \\ p_w \mathbf{q}_v + q_w \mathbf{p}_v + [\mathbf{p}_v]_\times \mathbf{q}_v \end{bmatrix} \tag{47}$$

计算

$$
\begin{aligned}
\mathbf{q}_{b_ib_{k+1}} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_{k+1}b'_{k+1}} \end{bmatrix} &= \mathbf{q}_{b_ib_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(\boldsymbol{\omega} - \delta\mathbf{b}^g_k)\delta t \end{bmatrix} \\
\begin{bmatrix} 1 \\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_{k+1}b'_{k+1}} \end{bmatrix} &\approx \begin{bmatrix} 1 \\ -\frac{1}{2}\boldsymbol{\omega}\delta t \end{bmatrix} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(\boldsymbol{\omega} - \delta\mathbf{b}^g_k)\delta t \end{bmatrix} \\
&= \begin{bmatrix} 1 + \frac{1}{4}\boldsymbol{\omega}^T(\boldsymbol{\omega} - \delta\mathbf{b}^g_k)\delta t^2 \\ -\frac{1}{2}\boldsymbol{\omega}\delta t + \frac{1}{2}(\boldsymbol{\omega} - \delta\mathbf{b}^g_k)\delta t + \frac{1}{4}[\boldsymbol{\omega}]_\times(\boldsymbol{\omega} - \delta\mathbf{b}^g_k)\delta t^2 \end{bmatrix} \\
\delta\boldsymbol{\theta}_{b_{k+1}b'_{k+1}} &= -\boldsymbol{\omega}\delta t + (\boldsymbol{\omega} - \delta\mathbf{b}^g_k)\delta t + \frac{1}{2}[\boldsymbol{\omega}]_\times(\boldsymbol{\omega} - \delta\mathbf{b}^g_k)\delta t^2
\end{aligned} \tag{48}
$$

疑问，上述过程中，利用等式两边四元数对应元素相等，需要四元数实部的$\frac{1}{4}\boldsymbol{\omega}^T(\boldsymbol{\omega} - \delta\mathbf{b}^g_k)\delta t^2 \approx 0$，问题是这个成立吗?

如果成立，那么雅克比得到

$$
\frac{\partial\delta\boldsymbol{\theta}_{b_{k+1}b'_{k+1}}}{\partial\delta\mathbf{b}^g_k} = -\mathbf{I}_{3\times3}\delta t - \frac{1}{2}[\boldsymbol{\omega}]_\times\delta t^2 \tag{49}
$$

4 预积分误差雅克比

上一节的雅克比主要是为了计算预积分测量值的协方差。迭代优化状态量的，误差$\mathbf{e}_B$对两个关键帧i,j的状态量$\mathbf{p},\mathbf{q},\mathbf{v},\mathbf{b}^a,\mathbf{b}^g$的雅克比也需要计算，因此这部分相关的误差雅克比进行推导。 对i,j时刻的状态量$\mathbf{p},\mathbf{q},\mathbf{v}$求导还是比较直观的，直接对误差公式进行计算就行。但是对i时刻的$\mathbf{b}^a_i,\mathbf{b}^g_i$求导就显得十分复杂，因为i时刻的bias卷进的预积分计算是通过迭代一步一步累计递推的，可以算但是太复杂。所以对于预积分量直接在i时刻的bias附近用一阶泰勒展开来近似，而不用真的去迭代计算。

$$
\begin{aligned}
\boldsymbol{\alpha}_{b_ib_j} &= \boldsymbol{\alpha}_{b_ib_j} + \mathbf{J}^{\alpha}_{b^a_i}\delta\mathbf{b}^a_i + \mathbf{J}^{\alpha}_{b^g_i}\delta\mathbf{b}^g_i \\
\boldsymbol{\beta}_{b_ib_j} &= \boldsymbol{\beta}_{b_ib_j} + \mathbf{J}^{\beta}_{b^a_i}\delta\mathbf{b}^a_i + \mathbf{J}^{\beta}_{b^g_i}\delta\mathbf{b}^g_i \\
\mathbf{q}_{b_ib_j} &= \mathbf{q}_{b_ib_j} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{J}^{q}_{b^g_i}\delta\mathbf{b}^g_i \end{bmatrix}
\end{aligned} \tag{50}
$$

其中 $\mathbf{J}^{\alpha}_{b^a_i} = \frac{\partial\boldsymbol{\alpha}_{b_ib_j}}{\partial\delta\mathbf{b}^a_i}, \mathbf{J}^{\alpha}_{b^g_i} = \frac{\partial\boldsymbol{\alpha}_{b_ib_j}}{\partial\delta\mathbf{b}^g_i}, \mathbf{J}^{\beta}_{b^a_i} = \frac{\partial\boldsymbol{\beta}_{b_ib_j}}{\partial\delta\mathbf{b}^a_i}, \mathbf{J}^{\beta}_{b^g_i} = \frac{\partial\boldsymbol{\beta}_{b_ib_j}}{\partial\delta\mathbf{b}^g_i}, \mathbf{J}^{q}_{b^g_i} = \frac{\mathbf{q}_{b_ib_j}}{den}$表示预积分量对$i$时刻的bias求导，这利用上一节的递推公式(35)可以得到。

由于$\mathbf{r}_p$和$\mathbf{r}_v$的误差形式很相近，对各状态量求导的雅克比形式也很相似，所以这里只对$\mathbf{r}_v$的推导进行详细介绍。这部分公式相应的代码主要对应imu_fatcor.h中雅克比的计算。

4.1 速度误差雅克比

(1) 对i时刻位移雅克比

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{p}_{b_i b_i'}} = \mathbf{0} \tag{51}$$

(2) 对i时刻旋转雅克比，对应代码公式jacobian[0]中$jacobian_pose_i.block < 3, 3 > (O_V, O_R)$

$$\begin{aligned}
\frac{\partial \mathbf{r}_v}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} &= \frac{\partial (\mathbf{q}_{wb_i} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta \boldsymbol{\theta}_{b_i b_i'} \end{bmatrix})^{-1} (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t)}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} \\
&= \frac{\partial (\mathbf{R}_{wb_i} \exp([\delta \boldsymbol{\theta}_{b_i b_i'}]_\times))^{-1} (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t)}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} \\
&= \frac{\partial \exp([-\delta \boldsymbol{\theta}_{b_i b_i'}]_\times) \mathbf{R}_{b_i w} (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t)}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} \\
&= \frac{\partial (\mathbf{I} - [\delta \boldsymbol{\theta}_{b_i b_i'}]_\times) \mathbf{R}_{b_i w} (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t)}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} \\
&= \frac{\partial (\mathbf{I} - [\delta \boldsymbol{\theta}_{b_i b_i'}]_\times) \mathbf{R}_{b_i w} (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t)}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} \\
&= \frac{\partial - [\delta \boldsymbol{\theta}_{b_i b_i'}]_\times \mathbf{R}_{b_i w} (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t)}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} \\
&= \frac{\partial [\mathbf{R}_{b_i w} (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t]_\times) \delta \boldsymbol{\theta}_{b_i b_i'}}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} \\
&= [\mathbf{R}_{b_i w} (\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t)]_\times
\end{aligned} \tag{52}$$

(3)对i时刻速度雅克比，对应jacobian[1]中$jacobian_speedbias_i.block < 3, 3 > (O_V, O_V - O_V)$

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{v}_i^w} = -\mathbf{R}_{b_i w} \tag{53}$$

(4)对i时刻的加速度bias雅克比，由于bias量只和预积分$\boldsymbol{\beta}$有关，所以

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{b}_i^a} = -\frac{\partial \boldsymbol{\beta}_{b_i b_j}}{\partial \delta \mathbf{b}_i^a} = -\mathbf{J}_{b_i^a}^{\beta} \tag{54}$$

由于$\boldsymbol{\beta}_{b_i b_j}$的离散计算形式采用的mid-point方法，所以这个雅克比和之前计算协方差矩阵时的雅克比是一样的，也就是可以将最后得到的j时刻相对i时刻的的雅克比公式(35)中$\mathbf{J}_j$取出对应的bias部分相关的导数作为这里的求解。

4.2 角度误差雅克比

(1)对i时刻姿态求导

$$
\begin{aligned}
\frac{\mathbf{r}_q}{\partial\delta\boldsymbol{\theta}_{b_ib_i'}} &= \frac{\partial 2[\mathbf{q}_{b_jb_i}\otimes(\mathbf{q}_{b_iw}\otimes\mathbf{q}_{wb_j})]_{xyz}}{\partial\delta\boldsymbol{\theta}_{b_ib_i'}} \\
&= \frac{\partial 2[\mathbf{q}^*_{b_ib_j}\otimes(\mathbf{q}_{wb_i}\otimes\begin{bmatrix}1\\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_ib_i'}\end{bmatrix})^*\otimes\mathbf{q}_{wb_j}]_{xyz}}{\partial\delta\boldsymbol{\theta}_{b_ib_i'}} \\
&= \frac{\partial -2[(\mathbf{q}^*_{b_ib_j}\otimes(\mathbf{q}_{wb_i}\otimes\begin{bmatrix}1\\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_ib_i'}\end{bmatrix})^*\otimes\mathbf{q}_{wb_j})^*]_{xyz}}{\partial\delta\boldsymbol{\theta}_{b_ib_i'}} \\
&= \frac{\partial -2[\mathbf{q}^*_{wb_j}\otimes(\mathbf{q}_{wb_i}\otimes\begin{bmatrix}1\\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_ib_i'}\end{bmatrix})\otimes\mathbf{q}_{b_ib_j}]_{xyz}}{\partial\delta\boldsymbol{\theta}_{b_ib_i'}} \\
&= -2\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\frac{\partial\mathbf{q}^*_{wb_j}\otimes(\mathbf{q}_{wb_i}\otimes\begin{bmatrix}1\\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_ib_i'}\end{bmatrix})\otimes\mathbf{q}_{b_ib_j}}{\partial\delta\boldsymbol{\theta}_{b_ib_i'}} \\
&= -2\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}\frac{\partial[\mathbf{q}^*_{wb_j}\otimes\mathbf{q}_{wb_i}]_L[\mathbf{q}_{b_ib_j}]_R\begin{bmatrix}1\\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_ib_i'}\end{bmatrix}}{\partial\delta\boldsymbol{\theta}_{b_ib_i'}} \\
&= -2\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}[\mathbf{q}^*_{wb_j}\otimes\mathbf{q}_{wb_i}]_L[\mathbf{q}_{b_ib_j}]_R\begin{bmatrix}\mathbf{0}\\ \frac{1}{2}\mathbf{I}\end{bmatrix}
\end{aligned}
\tag{55}
$$

(2)对j时刻姿态求导

$$
\begin{aligned}
\frac{\mathbf{r}_q}{\partial\delta\boldsymbol{\theta}_{b_jb_j'}} &= \frac{\partial 2[\mathbf{q}^*_{b_ib_j}\otimes\mathbf{q}^*_{wb_i}\otimes\mathbf{q}_{wb_j}\otimes\begin{bmatrix}1\\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_jb_j'}\end{bmatrix}]_{xyz}}{\partial\delta\boldsymbol{\theta}_{b_ib_i'}} \\
&= \frac{\partial 2[[\mathbf{q}^*_{b_ib_j}\otimes\mathbf{q}^*_{wb_i}\otimes\mathbf{q}_{wb_j}]_L\begin{bmatrix}1\\ \frac{1}{2}\delta\boldsymbol{\theta}_{b_jb_j'}\end{bmatrix}]_{xyz}}{\partial\delta\boldsymbol{\theta}_{b_ib_i'}} \\
&= -2\begin{bmatrix}\mathbf{0} & \mathbf{I}\end{bmatrix}[\mathbf{q}^*_{b_ib_j}\otimes\mathbf{q}^*_{wb_i}\otimes\mathbf{q}_{wb_j}]_L\begin{bmatrix}\mathbf{0}\\ \frac{1}{2}\mathbf{I}\end{bmatrix}
\end{aligned}
\tag{56}
$$

(2)对i时刻陀螺仪偏置b_i^g

$$
\begin{aligned}
\frac{\mathbf{r}_q}{\partial \delta \mathbf{b}_i^g} &= \frac{\partial 2[(\mathbf{q}_{b_i b_j} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{J}_{b_i^g}^q \delta \mathbf{b}_i^g \end{bmatrix})^* \otimes \mathbf{q}_{wb_i}^* \otimes \mathbf{q}_{wb_j}]_{xyz}}{\partial \delta \mathbf{b}_i^g} \\
&= \frac{\partial -2[((\mathbf{q}_{b_i b_j} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{J}_{b_i^g}^q \delta \mathbf{b}_i^g \end{bmatrix})^* \otimes \mathbf{q}_{wb_i}^* \otimes \mathbf{q}_{wb_j})^*]_{xyz}}{\partial \delta \mathbf{b}_i^g} \\
&= \frac{\partial -2[\mathbf{q}_{wb_j}^* \otimes \mathbf{q}_{wb_i} \otimes (\mathbf{q}_{b_i b_j} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\mathbf{J}_{b_i^g}^q \delta \mathbf{b}_i^g \end{bmatrix})]_{xyz}}{\partial \delta \mathbf{b}_i^g} \\
&= -2 \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} [\mathbf{q}_{wb_j}^* \otimes \mathbf{q}_{wb_i} \otimes \mathbf{q}_{b_i b_j}]_L \begin{bmatrix} \mathbf{0} \\ \frac{1}{2}\mathbf{J}_{b_i^g}^q \end{bmatrix}
\end{aligned} \tag{57}
$$

5 视觉测量

沈老师的代码中，用针孔模型将相机坐标系下的三维空间点$\begin{bmatrix} x \\ y \\ z \end{bmatrix}$投影到**归一化图像平面**进行坐标的误差计算，而不是图像平面。假设该特征点在归一化图像平面的坐标为$\begin{bmatrix} u \\ v \end{bmatrix}$，重投影误差函数定义为

$$
\mathbf{r}_c = \begin{bmatrix} \frac{x}{z} - u \\ \frac{y}{z} - v \end{bmatrix} \tag{58}
$$

利用特征点的逆深度λ可以把点特征从归一化图像平面的二维坐标经过反投影，得到其在相机坐标系下的三维坐标

$$
\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{59}
$$

一般特征点在第i个关键帧初始化，需要投影到第j个图像帧上去计算重投影误差，这个过程用齐次坐标变换，需要经过如下运算：

$$\begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \\ 1 \end{bmatrix} = \mathbf{T}_{bc}^{-1}\mathbf{T}_{wb_j}^{-1}\mathbf{T}_{wb_i}\mathbf{T}_{bc}\begin{bmatrix} \frac{1}{\lambda}u_{c_i} \\ \frac{1}{\lambda}v_{c_i} \\ \frac{1}{\lambda} \\ 1 \end{bmatrix} \tag{60}$$

很容易拆开成三维坐标形式：

$$\mathbf{f}_{c_j} = \begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \end{bmatrix} = \mathbf{R}_{bc}^T\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i}\mathbf{R}_{bc}\frac{1}{\lambda}\begin{bmatrix} u_{c_i} \\ v_{c_i} \\ 1 \end{bmatrix} + \mathbf{R}_{bc}^T(\mathbf{R}_{wb_j}^T((\mathbf{R}_{wb_i}\mathbf{p}_{bc} + \mathbf{p}_{wb_i}) - \mathbf{p}_{wb_j}) - \mathbf{p}_{bc}) \tag{61}$$

再推导各类雅克比之前，为了简化公式，先定义如下变量：

$$\begin{aligned} \mathbf{f}_{b_i} &= \mathbf{R}_{bc}\mathbf{f}_{c_i} + \mathbf{p}_{bc} \\ \mathbf{f}_{w} &= \mathbf{R}_{wb_i}\mathbf{f}_{b_i} + \mathbf{p}_{wb_i} \\ \mathbf{f}_{b_j} &= \mathbf{R}_{wb_j}^T(\mathbf{f}_{c_i} - \mathbf{p}_{bc}) \end{aligned} \tag{62}$$

5.1 视觉误差雅克比

误差方程：

$$\mathbf{r}_c = \begin{bmatrix} \frac{x_{c_j}}{z_{c_j}} - u_{c_j} \\ \frac{y_{c_j}}{z_{c_j}} - v_{c_j} \end{bmatrix} \tag{63}$$

雅克比为误差对两个时刻的状态量，外参数，以及逆深度求导：

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{r}_c}{\partial \begin{bmatrix} \delta\mathbf{p}_{b_i b_i'} \\ \delta\boldsymbol{\theta}_{b_i b_i'} \end{bmatrix}} & \frac{\partial \mathbf{r}_c}{\partial \begin{bmatrix} \delta\mathbf{p}_{b_j b_j'} \\ \delta\boldsymbol{\theta}_{b_j b_j'} \end{bmatrix}} & \frac{\partial \mathbf{r}_c}{\partial \begin{bmatrix} \delta\mathbf{p}_{cc'} \\ \delta\boldsymbol{\theta}_{cc'} \end{bmatrix}} & \frac{\partial \mathbf{r}_c}{\partial \delta\lambda} \end{bmatrix} \tag{64}$$

根据链式法则，雅克比的计算可以分两步走，第一步误差对$\mathbf{f}_{c_j}$求导，第二步$\mathbf{f}_{c_j}$对各状态量求导。

$$\frac{\partial \mathbf{r}_c}{\partial \mathbf{f}_{c_j}} = \begin{bmatrix} \frac{1}{z_{c_j}} & 0 & -\frac{x_{c_j}}{z_{c_j}^2} \\ 0 & \frac{1}{z_{c_j}} & -\frac{y_{c_j}}{z_{c_j}^2} \end{bmatrix} \tag{65}$$

5.1.1 对i时刻的状态量求导

对位移求导，这个还是比较简单的，直接写出如下：

$$\frac{\partial \mathbf{f}_{c_j}}{\partial \delta \mathbf{p}_{b_i b_i'}} = \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \tag{66}$$

对角度增量求导

$$\mathbf{f}_{c_j} = \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \mathbf{R}_{wb_i} \mathbf{R}_{bc} \mathbf{f}_{c_i} + \mathbf{R}_{bc}^T (\mathbf{R}_{wb_j}^T ((\mathbf{R}_{wb_i} \mathbf{p}_{bc} + \mathbf{p}_{wb_i}) - \mathbf{p}_{wb_j}) - \mathbf{p}_{bc}) \tag{67}$$

上面公式看着吓人，但是和i时刻角度相关的量并不多，下面为了简化，我直接丢弃了不相关的部分

$$\begin{aligned} \mathbf{f}_{c_j} &= \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \mathbf{R}_{wb_i} \mathbf{R}_{bc} \mathbf{f}_{c_i} + \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \mathbf{R}_{wb_i} \mathbf{p}_{bc} + (...) \\ &= \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \mathbf{R}_{wb_i} (\mathbf{R}_{bc} \mathbf{f}_{c_i} + \mathbf{p}_{bc}) + (...) \\ &= \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \mathbf{R}_{wb_i} \mathbf{f}_{b_i} + (...) \end{aligned} \tag{68}$$

雅克比为

$$\begin{aligned} \frac{\partial \mathbf{f}_{c_j}}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} &= \frac{\partial \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \mathbf{R}_{wb_i} (\mathbf{I} + [\delta \boldsymbol{\theta}_{b_i b_i'}]_\times) \mathbf{f}_{b_i}}{\partial \delta \boldsymbol{\theta}_{b_i b_i'}} \\ &= -\mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \mathbf{R}_{wb_i} [\mathbf{f}_{b_i}]_\times \end{aligned} \tag{69}$$

5.1.2 对j时刻的状态量求导

对位移求导

$$\frac{\partial \mathbf{f}_{c_j}}{\partial \delta \mathbf{p}_{b_j b_j'}} = -\mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \tag{70}$$

对角度增量求导，同上面的操作，也简化一下公式

$$\begin{aligned} \mathbf{f}_{c_j} &= \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T \mathbf{R}_{wb_i} \mathbf{R}_{bc} \mathbf{f}_{c_i} + \mathbf{R}_{bc}^T (\mathbf{R}_{wb_j}^T ((\mathbf{R}_{wb_i} \mathbf{p}_{bc} + \mathbf{p}_{wb_i}) - \mathbf{p}_{wb_j}) - \mathbf{p}_{bc}) \\ &= \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T (\mathbf{R}_{wb_i} (\mathbf{R}_{bc} \mathbf{f}_{c_i} + \mathbf{p}_{bc}) + \mathbf{p}_{wb_i} - \mathbf{p}_{wb_j}) + (...) \\ &= \mathbf{R}_{bc}^T \mathbf{R}_{wb_j}^T (\mathbf{f}_w - \mathbf{p}_{wb_j}) + (...) \end{aligned} \tag{71}$$

雅克比为

$$\begin{aligned} \frac{\partial \mathbf{f}_{c_j}}{\partial \delta \boldsymbol{\theta}_{b_j b_j'}} &= \frac{\partial \mathbf{R}_{bc}^T (\mathbf{I} - [\delta \boldsymbol{\theta}_{b_j b_j'}]_\times) \mathbf{R}_{wb_j}^T (\mathbf{f}_w - \mathbf{p}_{wb_j})}{\partial \delta \boldsymbol{\theta}_{b_j b_j'}} \\ &= \frac{\partial \mathbf{R}_{bc}^T (\mathbf{I} - [\delta \boldsymbol{\theta}_{b_j b_j'}]_\times) \mathbf{f}_{b_j}}{\partial \delta \boldsymbol{\theta}_{b_j b_j'}} \\ &= \mathbf{R}_{bc}^T [\mathbf{f}_{b_j}]_\times \end{aligned} \tag{72}$$

5.1.3 对imu和相机之间的外参数状态量求导

对位移求导

$$\frac{\partial \mathbf{f}_{c_j}}{\partial \delta \mathbf{p}_{cc'}} = \mathbf{R}_{bc}^T(\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i} - \mathbf{I}_{3\times3}) \tag{73}$$

对角度增量求导，由于$\mathbf{f}_{c_j}$都和$\mathbf{R}_{bc}$有关，并且比较复杂，所以这次分两部分求导，不然公式太长了

$$\begin{aligned}\mathbf{f}_{c_j} &= \mathbf{R}_{bc}^T\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i}\mathbf{R}_{bc}\mathbf{f}_{c_i} + \mathbf{R}_{bc}^T(\mathbf{R}_{wb_j}^T((\mathbf{R}_{wb_i}\mathbf{p}_{bc} + \mathbf{p}_{wb_i}) - \mathbf{p}_{wb_j}) - \mathbf{p}_{bc}) \\ &= \mathbf{f}_{c_j}^1 + \mathbf{f}_{c_j}^2\end{aligned} \tag{74}$$

两部分雅克比分别为

$$\begin{aligned}\frac{\partial \mathbf{f}_{c_j}^1}{\partial \delta \boldsymbol{\theta}_{cc'}} &= \frac{\partial (\mathbf{I} - [\delta\boldsymbol{\theta}_{cc'}]_\times)\mathbf{R}_{bc}^T\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i}\mathbf{R}_{bc}(\mathbf{I} + [\delta\boldsymbol{\theta}_{cc'}]_\times)\mathbf{f}_{c_i}}{\partial \delta\boldsymbol{\theta}_{cc'}} \\ &= \frac{\partial \mathbf{R}_{bc}^T\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i}\mathbf{R}_{bc}[\delta\boldsymbol{\theta}_{cc'}]_\times\mathbf{f}_{c_i} - [\delta\boldsymbol{\theta}_{cc'}]_\times\mathbf{R}_{bc}^T\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i}\mathbf{R}_{bc}\mathbf{f}_{c_i} + o^2(\delta\boldsymbol{\theta}_{cc'}) + (...)}{\partial \delta\boldsymbol{\theta}_{cc'}} \\ &= -\mathbf{R}_{bc}^T\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i}\mathbf{R}_{bc}[\mathbf{f}_{c_i}]_\times + [\mathbf{R}_{bc}^T\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i}\mathbf{R}_{bc}\mathbf{f}_{c_i}]_\times \\ \frac{\partial \mathbf{f}_{c_j}^2}{\partial \delta \boldsymbol{\theta}_{cc'}} &= \frac{(\mathbf{I} - [\delta\boldsymbol{\theta}_{cc'}]_\times)\mathbf{R}_{bc}^T(\mathbf{R}_{wb_j}^T((\mathbf{R}_{wb_i}\mathbf{p}_{bc} + \mathbf{p}_{wb_i}) - \mathbf{p}_{wb_j}) - \mathbf{p}_{bc})}{\partial \delta\boldsymbol{\theta}_{cc'}} \\ &= [\mathbf{R}_{bc}^T(\mathbf{R}_{wb_j}^T((\mathbf{R}_{wb_i}\mathbf{p}_{bc} + \mathbf{p}_{wb_i}) - \mathbf{p}_{wb_j}) - \mathbf{p}_{bc})]_\times\end{aligned} \tag{75}$$

两个雅克比相加就是最后的结果了。

5.1.4 对特征逆深度的求导

$$\begin{aligned}\frac{\partial \mathbf{f}_{c_j}}{\partial \delta \lambda} &= \frac{\partial \mathbf{f}_{c_j}}{\partial \mathbf{f}_{c_i}}\frac{\partial \mathbf{f}_{c_i}}{\partial \delta \lambda} \\ &= \mathbf{R}_{bc}^T\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i}\mathbf{R}_{bc}(-\frac{1}{\lambda^2}\begin{bmatrix}u_{c_i}\\ v_{c_i}\\ 1\end{bmatrix}) \\ &= -\frac{1}{\lambda}\mathbf{R}_{bc}^T\mathbf{R}_{wb_j}^T\mathbf{R}_{wb_i}\mathbf{R}_{bc}\mathbf{f}_{c_i}\end{aligned} \tag{76}$$

5.1.5 扩展：沈老师他们提出的球面切平面重投影误差雅克比

上述重投影误差是将预测的特征点坐标$\mathbf{f}_{c_j}$直接除以z轴坐标，将其投影到归一化图像平面进行的误差计算。而球面切平面的计算过程稍微不一样，在检测到的特征点方向向量$\hat{\mathbf{f}}_{c_j} = \begin{bmatrix}u_{c_j}\\ v_{c_j}\\ 1\end{bmatrix}$的单元球面处构建切向平面。 首先计算，

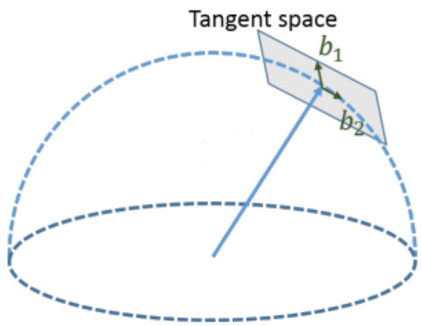

Figure 1: 在特征点的方向向量$\mathbf{f}'$的单位球面切平面上找到一个基底b_1, b_2，方向向量之间的误差$\mathbf{f} - \mathbf{f}'$投影到这个二维切平面上进行计算。

两个方向向量在球平面的误差

$$
\begin{aligned}
\mathbf{r}_{sphere} &= \frac{\mathbf{f}_{c_j}}{||\mathbf{f}_{c_j}||} - \frac{\hat{\mathbf{f}}_{c_j}}{||\hat{\mathbf{f}}_{c_j}||} \\
&= \frac{1}{\sqrt{x_{c_j}^2 + y_{c_j}^2 + z_{c_j}^2}} \begin{bmatrix} x_{c_j} \\ y_{c_j} \\ z_{c_j} \end{bmatrix} - \frac{\hat{\mathbf{f}}_{c_j}}{||\hat{\mathbf{f}}_{c_j}||}
\end{aligned} \tag{77}
$$

其实得到的就是两个向量的端点相邻组成的向量，因为图像误差是二维的，所以把这个三维的向量投影到切平面上，用基底b_1, b_2表示，基底b是一个三维列向量

$$
\mathbf{r}_{tangent} = \begin{bmatrix} b_1^T \\ b_2^T \end{bmatrix} \mathbf{r}_{sphere} \tag{78}
$$

两个基底是垂直于特征点方向向量$\hat{\mathbf{f}}_{c_j}$的，作者使用施密特正交法得到。

残差对各状态变量的求导也和上面一样，用链式法则可以分成两步走。先计算$\frac{\partial \mathbf{r}_{tangent}}{\partial \mathbf{f}_{c_j}}$，再计算$\mathbf{f}_{c_j}$对各变量的导数，后面这一部分的导数和之前在重投影误差计算雅克比一节中已经计算。因此只要计算前一部分。而这部分导数的计算利用简单的数学知识就能求解，这里不再赘述。

References

[1] J. Sola, "Quaternion kinematics for the error-state kalman filter," 2017.

[2] Y.-B. Jia, "Quaternion and rotation," 2016.

[3] N. Trawny and S. I. Roumeliotis, "Indirect kalman filter for 3d attitude estimation," 2005.