

问题1

课程历程中的右扰动雅可比 $J_r(\phi)^{-1}$ 没看懂如何推导出来的。老师能解答一下吗？查阅了相关资料也对不上公式

```
static Eigen::Matrix3d JacobianRInv(const Eigen::Vector3d &w) {
    Eigen::Matrix3d J_r_inv = Eigen::Matrix3d::Identity();

    double theta = w.norm();

    if (theta > 1e-5) {
        Eigen::Vector3d k = w.normalized();
        Eigen::Matrix3d K = Sophus::SO3d::hat(k);

        J_r_inv = J_r_inv
            + 0.5 * K
            + (1.0 - (1.0 + std::cos(theta)) * theta / (2.0 * std::sin(theta))) * K * K;
    }

    return J_r_inv;
}
```

问题2

进行舒尔补构建b矩阵时为什么不是 $b = -J^\top f$ ？看了vins-mono的程序它是没有负号的，而ppt中的b矩阵是有负号的

```
// 节选自vins-mono代码
b.segment(idx_i, size_i) += jacobian_i.transpose() * it->residuals;

// 节选自葛助教github参考代码
// a. b_m:
b_.block<15,1>(INDEX_M,0) += J_m.transpose() * residuals;
// a. b_r:
b_.block<15,1>(INDEX_R,0) += J_r.transpose() * residuals;
```

基于kitti的实现原理

1. 基于地图定位的滑动窗口模型

5) 完整模型

上述过程用公式可表示为：

$$\underbrace{J^\top \Sigma J}_H \delta x = - \underbrace{J^\top \Sigma r}_b$$

其中

$$r = \begin{bmatrix} r_{Y0} \\ r_{Y1} \\ r_{Y2} \\ r_{L0} \\ r_{L1} \\ r_{M0} \\ r_{M1} \end{bmatrix}$$

$$J = \frac{\partial r}{\partial \delta x} = \begin{bmatrix} \frac{\partial r_{Y0}}{\partial \delta x} \\ \frac{\partial r_{Y1}}{\partial \delta x} \\ \frac{\partial r_{Y2}}{\partial \delta x} \\ \frac{\partial r_{L0}}{\partial \delta x} \\ \frac{\partial r_{L1}}{\partial \delta x} \\ \frac{\partial r_{M0}}{\partial \delta x} \\ \frac{\partial r_{M1}}{\partial \delta x} \end{bmatrix} = \begin{bmatrix} J_1 \\ J_2 \\ J_3 \\ J_4 \\ J_5 \\ J_6 \\ J_7 \end{bmatrix}$$
$$J^\top = [J_1^\top \ J_2^\top \ J_3^\top \ J_4^\top \ J_5^\top \ J_6^\top \ J_7^\top]$$

矩阵乘法写成累加形式为：

$$\sum_{i=1}^7 J_i^\top \Sigma_i J_i \delta x = - \sum_{i=1}^7 J_i^\top \Sigma_i r_i$$

此累加过程，即对应前面可视化时各矩阵叠加。

问题3

vins-mono中使用舒尔补后的H, b矩阵反算出先验因子的residual和jacobian是什么数学原理? 具体是什么公式? 哪里可以找到相关论文或资料? 感觉这个以后也可以加到课程内容里, 因为做非线性优化更需要知道jacobian和residual是怎么来的。

```
A = Arr - Arm * Amm_inv * Amr;  
b = brr - Arm * Amm_inv * bmm;  
  
Eigen::SelfAdjointEigenSolver<Eigen::MatrixXd> saes2(A);  
Eigen::VectorXd S = Eigen::VectorXd((saes2.eigenvalues().array() > eps).select(saes2.eigenvalues().array(), 0));  
Eigen::VectorXd S_inv = Eigen::VectorXd((saes2.eigenvalues().array() > eps).select(saes2.eigenvalues().array().inverse(), 0));  
  
Eigen::VectorXd S_sqrt = S.cwiseSqrt();  
Eigen::VectorXd S_inv_sqrt = S_inv.cwiseSqrt();  
  
linearized_jacobians = S_sqrt.asDiagonal() * saes2.eigenvectors().transpose();  
linearized_residuals = S_inv_sqrt.asDiagonal() * saes2.eigenvectors().transpose() * b;
```

问题4

程序预积分的jacobian是使用旋转矩阵求导, 姿态的残差对陀螺仪bias的求导是怎么求的? 我求出来的结果与葛助教github上代码不同, 下面是我的求导过程 (r_g应该改为r_q):

1. 扰动是这样加的吗?
2. 求导结果有没有错误?

The image shows a handwritten derivation of the Jacobian for attitude residual with respect to gyro bias. The derivation starts with the expression for the residual and proceeds through several steps, including adding a perturbation, using the chain rule, and finally applying the BCH approximation.

$$\begin{aligned} \frac{\partial r_g}{\partial \delta b_g} &= \frac{\partial \ln [y_{b_i b_j}^T \cdot R_{b_i}^w{}^T \cdot R_{b_j}^w]}{\partial \delta b_g} \\ &\stackrel{\text{加扰动}}{=} \frac{\partial \ln [y_{b_i b_j}^T \cdot (R_{b_i}^w \cdot \exp([J \delta b_g]_x)){}^T \cdot R_{b_j}^w]}{\partial \delta b_g} \\ &= \frac{\partial \ln [y_{b_i b_j}^T \cdot \exp(-[J \delta b_g]_x) \cdot R_{b_i}^w{}^T \cdot R_{b_j}^w]}{\partial \delta b_g} \\ &\stackrel{\text{用伴随}}{=} \frac{\partial \ln [y_{b_i b_j}^T \cdot R_{b_i}^w{}^T \cdot \exp(-[R_{b_i}^w \cdot J \cdot \delta b_g]_x) \cdot R_{b_j}^w]}{\partial \delta b_g} \\ &\stackrel{\text{用伴随}}{=} \frac{\partial \ln [y_{b_i b_j}^T \cdot R_{b_i}^w{}^T \cdot R_{b_j}^w \cdot \exp(-[R_{b_j}^w{}^T \cdot R_{b_i}^w \cdot J \cdot \delta b_g]_x)]}{\partial \delta b_g} \\ &\stackrel{\text{BCH}}{=} -J_r^T (\ln [y_{b_i b_j}^T \cdot R_{b_i}^w{}^T \cdot R_{b_j}^w]) \cdot R_{b_j}^w{}^T \cdot R_{b_i}^w \cdot J \end{aligned}$$

其中, $J = J_{b_i b_j}^{y_{b_i b_j}}$, $y_{b_i b_j}$ 为陀螺仪预积分。

问题5

1. 课程例程中，使用scancontext的结果作为地图匹配的约束（一元约束），使用两帧与地图匹配的结果作为里程计约束（即 $T_{b_2}^{b_1} = T_{b_1}^w \top T_{b_2}^w$ ）这是ppt上附加题的意思？
2. 我个人理解是：与地图匹配的结果作为一元约束，再跑个loam的scan2scan作为里程计约束，这样是否正确？
3. 里程计约束是否可以跑scan2map的方法，比如前四章程序里面的那种基于icp的scan2map里程计

问题6

1. 老师在课上说有些组合导航能输出bias，但我接触过的几款国内的组合导航都没输出bias，这时我们有没有必要写一个程序专门估计初始的bias？一些开源飞控（px4）也没估初始的bias也很准。
2. 使用vins-mono的初始化时是不是要绕八字才能初始化好？我在kitti上跑LIO-Mapping基本上是初始化不成功的，计算初始化成功建图效果也很差。无人驾驶中是怎么初始化的？如果是无感初始化的话用户也只是“走一段直线拐一个弯，再走一段直线再拐个弯而已”，远远没有人为绕八字激励那么大呀
3. 最近看小鹏p5的城市NGP辅助驾驶demo，它能完成从地下车库出发到目的地，它的两个激光雷达应该只是用来避障，没有用来定位的吧？