公式推导：

$$\begin{bmatrix} r_p \\ r_q \\ r_v \\ r_{ba} \\ r_{bg} \end{bmatrix} = \begin{bmatrix} q_{wb_i}^{*}(p_{wb_j} - p_{wb_i} - v_i^w \Delta t + \frac{1}{2} g^w \Delta t^2) - \alpha_{b_i b_j} \\ 2[q_{b_i b_j}^{*} \otimes (q_{wb_i}^{*} \otimes q_{wb_j})]_{xyz} \\ q_{wb_i}^{*}(v_j^w - v_i^w + g^w \Delta t) - \beta_{b_i b_j} \\ b_j^a - b_i^a \\ b_j^g - b_i^g \end{bmatrix}$$

$$\boxed{\frac{\partial r_{p_i}}{\partial p_{wb_i}} = \left[-q_{wb_i}^{*}\right]} \qquad \frac{\partial r_{p_i}}{\partial p_{wb_j}} = \left[q_{wb_i}^{*}\right]$$

$$\frac{\partial r_{p_i}}{\partial \theta_{wb_i}} = \frac{\partial \left[q_{wb_i} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\theta_{b_i} \end{bmatrix}\right]^{*}(p_{wb_j} - p_{wb_i} - v_i^w \Delta t + \frac{1}{2} g^w \Delta t^2)}{\partial \delta\theta_{b_i}}$$

第i个

第i个节点

$$= \frac{\partial (R_{wb_i} \exp(\delta\theta_{b_i}^{\wedge}))^{-1}(p_{wb_j} - p_{wb_i} - v_i^w \Delta t + \frac{1}{2} g^w \Delta t^2)}{\partial \delta\theta_{b_i}}$$

$$= \frac{\partial \exp(-\delta\theta_{b_i}^{\wedge}) R_{wb_i}^{-1}(p_{wb_j} - p_{wb_i} - v_i^w \Delta t + \frac{1}{2} g^w \Delta t^2)}{\partial \delta\theta_{b_i}}$$

泰勒展开(一阶)

$$= \frac{\partial (I - \delta\theta_{b_i}^{\wedge}) R_{wb_i}^{-1}(p_{wb_j} - p_{wb_i} - v_i^w \Delta t + \frac{1}{2} g^w \Delta t^2)}{\partial \delta\theta_{b_i}}$$

$$\boxed{= R_{wb_i}^{-1}(p_{wb_j} - p_{wb_i} - v_i^w \Delta t + \frac{1}{2} g^w \Delta t^2)^{\wedge}}$$

$$\frac{\partial r_{p_i}}{\partial v_i} = -q_{wb_i}^{*} \Delta t = -R_{wb_i}^{-1} \Delta t = -R_{b_i w} \Delta t$$

$$\frac{\partial r_{p_i}}{\partial b_{a_i}} = \frac{\partial(-\alpha_{b_i b_j})}{\partial b_{a_i}} = \frac{\partial -(\bar{\alpha}_{b_i b_j} + J_{b_i^a}^{\alpha} \delta b_i^a + J_{b_i^g}^{\alpha} \delta b_i^g)}{\partial b_{a_i}}$$

$$= -J_{b_i^a}^{\alpha}$$

$\frac{\partial r_{p_i}}{\partial b_{g_i}}$ 同上：$-J_{b_i^g}^{\alpha}$

$$\frac{\partial r_{q_i}}{\partial p_i} = 0$$

$$\frac{\partial r_{q_i}}{\partial q_i} = 2\frac{\partial \left[q_{b_j b_i} \otimes \left(q_{b_i w} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\theta_{b_i b_i'} \end{bmatrix}\right)^{*} \otimes q_{wb_j}\right]_{xyz}}{\partial \delta\theta_{b_i b_i'}}$$

$$= 2\frac{\partial\left[q_{wb_j}^{*}\otimes\left(q_{b_i w}\otimes\begin{bmatrix}1\\ \frac{1}{2}\delta\theta_{b_i b_i'}\end{bmatrix}\right)\otimes q_{b_j b_i}^{*}\right]_{xyz}}{\partial\delta\theta_{b_i b_i'}}$$

$$= -2\begin{bmatrix}0 & I\end{bmatrix}\left[q_{wb_j}^{*}\otimes q_{wb_i}\right]_L\left[q_{b_j b_i}^{*}\right]_R\begin{bmatrix}0\\ \frac{1}{2}I\end{bmatrix}.$$

$$\frac{\partial r_{q_i}}{\partial v_i} = 0 \qquad \frac{\partial r_{q_i}}{\partial b_{a_i}} = 0$$

$$\frac{\partial r_{q_i}}{\partial b_{g_i}} = 2\frac{\partial\left[\left(q_{b_i b_j}\begin{bmatrix}1\\ \frac{1}{2}J^{q}_{b_i^g}\delta b_i^g\end{bmatrix}\right)^{*}\otimes\left(q_{wb_i}^{*}\otimes q_{wb_j}\right)\right]_{xyz}}{\partial\delta b_i^g}$$

$$= -2\begin{bmatrix}0 & I\end{bmatrix}\frac{\partial\left(\begin{bmatrix}1\\ \frac{1}{2}J^{q}_{b_i^g}\delta b_i^g\end{bmatrix}^{*}\bar{q}_{b_i b_j}^{*}\otimes q_{wb_i}^{*}\otimes q_{wb_j}\right)^{*}}{\partial\delta b_i^g}$$

$$= -2\begin{bmatrix}0 & I\end{bmatrix}\left[q_{wb_j}^{*}\otimes q_{wb_i}\otimes q_{b_i b_j}\right]_L\begin{bmatrix}0\\ \frac{1}{2}J^{q}_{b_i^g}\end{bmatrix}$$

$$\frac{\partial r_{v_i}}{\partial p_i} = 0$$

$$\frac{\partial r_{v_i}}{\partial q_i} = \frac{\partial\left(R_{wb_i}\exp[\delta\theta_{b_i b_i'}]\right)^{-1}\left(v_j^w - v_i^w + g^w\Delta t\right)}{\partial\delta\theta_{b_i b_i'}}$$

$$= \frac{\left(I - [\delta\theta_{b_i b_i'}]^{\wedge}\right)R_{wb_i}^{-1}\left(v_j^w - v_i^w + g^w\Delta t\right)}{\partial\delta\theta_{b_i b_i'}}$$

$$= \left[R_{wb_i}^{-1}\left(v_j^w - v_i^w + g^w\Delta t\right)\right]^{\wedge}$$

$$\frac{\partial r_{v_j}}{\partial v_i} = -R_{wb_i}^{*}$$

$$\frac{\partial r_{v_j}}{\partial b_{a_i}} = -\frac{\partial\beta_{b_i b_j}}{\partial b_{a_i}}$$

$$= -\frac{\partial\bar{\beta}_{b_i b_j} + J^{\beta}_{b_i^a}\delta b_i^a + J^{\beta}_{b_i^g}\delta b_i^g}{\partial b_{a_i}}$$

$$= -J^{\beta}_{b_i^a}$$

$$\frac{\partial r_{v_i}}{\partial b_{g_i}} = -\frac{\partial \beta_{b_i b_j}}{\partial b_{g_i}}$$

$$= -\frac{\partial \bar{\beta}_{b_i b_j} + J^{\beta}_{b_i^a}\,\delta b_i^a + J^{\beta}_{b_i^g}\,\delta b_i^g}{\partial b_{g_i}}$$

$$= -J^{\beta_g}_{b_i}$$

$$\frac{\partial r_{ba_i}}{\partial p_i} = 0,\quad \frac{\partial r_{ba_i}}{\partial q_i} = 0,\quad \frac{\partial r_{ba_i}}{\partial v_i} = 0,\quad \frac{\partial r_{ba_i}}{\partial_{ba_i}} = -I$$

$$\frac{\partial r_{ba_i}}{\partial b_{g_i}} = 0$$

$$\frac{\partial b_{gi}}{\partial p_i} = 0, \quad \frac{\partial b_{gi}}{\partial q_i} = 0, \quad \frac{\partial b_{gi}}{\partial v_i} = 0, \quad \frac{\partial r_{bgi}}{\partial b_{ai}} = 0.$$

$$\frac{\partial r_{bgi}}{\partial b_{gi}} = -I$$

$$\frac{\partial r_{pi}}{\partial p_j} = R^{-1}_{wb_i}, \quad \frac{\partial r_{pi}}{\partial q_j} = 0, \quad \frac{\partial r_{pi}}{\partial v_j} = 0.$$

$$\frac{\partial r_{pi}}{\partial b_{aj}} = 0, \quad \frac{\partial r_{pi}}{\partial b_{gj}} = 0$$

$$\frac{\partial r_{qi}}{\partial p_j} = 0, \quad \frac{\partial r_{qi}}{\partial q_j} = 2\frac{\partial}{\partial \delta\theta_{b_j b_j'}}\left[q^*_{b_i b_j} \otimes \left(q^*_{wb_i} \otimes q_{wb_j} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta\theta_{b_j b_j'} \end{bmatrix}\right)\right]$$

$$= 2\begin{bmatrix} 0 & I \end{bmatrix}\left[q^*_{b_i b_j} \otimes q^*_{wb_i} \otimes q_{wb_j}\right]_L \begin{bmatrix} 0 \\ \frac{1}{2}I \end{bmatrix}$$

$$\frac{\partial r_{qi}}{\partial v_j} = 0, \quad \frac{\partial r_{qi}}{\partial b_{aj}} = \frac{\partial r_{qi}}{\partial b_{gj}} = 0$$

$$\frac{\partial r_{v_i}}{\partial p_j} = \frac{\partial r_{v_i}}{\partial q_j} = \frac{\partial r_{v_i}}{\partial b_{a_j}} = \frac{\partial r_{v_i}}{\partial b_{g_j}} = 0.$$

$$\frac{\partial r_{v_i}}{\partial v_j} = R_{wb_i}^{T}$$

$$\frac{\partial r_{ba_i}}{\partial p_j} = \frac{\partial r_{ba_i}}{\partial q_j} = \frac{\partial r_{ba_i}}{\partial v_j} = \frac{\partial r_{ba_i}}{\partial b_{g_j}} = 0.$$

$$\frac{\partial r_{ba_i}}{\partial b_{a_j}} = I$$

$$\frac{\partial r_{bg_i}}{\partial p_j} = \frac{\partial r_{bg_i}}{\partial q_j} = \frac{\partial r_{bg_i}}{\partial v_j} = \frac{\partial r_{bg_i}}{\partial b_{a_j}} = 0.$$

$$\frac{\partial r_{bg_i}}{\partial b_{g_j}} = I$$

补全代码：

UpdateState: 中值法更新最优先验，然后F和B相对应的值补进去就行了

```
void IMUPreIntegrator::UpdateState(void) {
...
    //
    // TODO: a. update mean:
    //
    // 1. get w_mid:
    w_mid = 0.5 * (prev_w + curr_w);
    // 2. update relative orientation, so3:
    prev_theta_ij = state.theta_ij_;
    d_theta_ij = Sophus::SO3d::exp(w_mid*T);
    curr_theta_ij = prev_theta_ij * d_theta_ij;
    state.theta_ij_ = curr_theta_ij;
    // 3. get a_mid:
    a_mid = 0.5 * (prev_a + curr_a);
    // 4. update relative translation:
    state.alpha_ij_ += state.beta_ij_*T + 0.5*a_mid*T*T;
    // 5. update relative velocity:
    state.beta_ij_ += a_mid*T;
```

```
    //
    // TODO: b. update covariance:
    //
    // 1. intermediate results:
    dR_inv = d_theta_ij.inverse().matrix();
    prev_R = prev_theta_ij.matrix();
    curr_R = curr_theta_ij.matrix();
    prev_R_a_hat = prev_R * Sophus::SO3d::hat(prev_a);
    curr_R_a_hat = curr_R * Sophus::SO3d::hat(curr_a);
    //
    // TODO: 2. set up F:
    //
    // F12 & F32:
    F_ = MatrixF::Identity();
    F_.block<3,3>(INDEX_ALPHA, INDEX_THETA) = -0.25*T*T * (prev_R_a_hat +
curr_R_a_hat*(Eigen::Matrix3d::Identity() - Sophus::SO3d::hat(w_mid)*T));
    F_.block<3,3>(INDEX_BETA, INDEX_THETA) = -0.5*T * (prev_R_a_hat +
curr_R_a_hat*(Eigen::Matrix3d::Identity() - Sophus::SO3d::hat(w_mid)*T));
    // F14 & F34:
    F_.block<3,3>(INDEX_ALPHA, INDEX_B_A) = -0.25 * (prev_R + curr_R) * T*T;
    F_.block<3,3>(INDEX_BETA, INDEX_B_A) = -0.5 * (prev_R + curr_R) * T;
    // F15 & F35:
    F_.block<3,3>(INDEX_ALPHA, INDEX_B_G) = 0.25*T*T*T * curr_R_a_hat;
    F_.block<3,3>(INDEX_BETA, INDEX_B_G) = 0.5*T*T * curr_R_a_hat;
    // F22:
    F_.block<3,3>(INDEX_THETA, INDEX_THETA) = Eigen::Matrix3d::Identity() -
Sophus::SO3d::hat(w_mid)*T;
    //
    // TODO: 3. set up G:
    //
    // G11 & G31:
    B_ = MatrixB::Zero();
    B_.block<3,3>(INDEX_ALPHA, INDEX_M_ACC_PREV) = 0.25*T*T * prev_R;
    B_.block<3,3>(INDEX_BETA, INDEX_M_ACC_PREV) = 0.5*T * prev_R;
    // G12 & G22 & G32:
    B_.block<3,3>(INDEX_ALPHA, INDEX_M_GYR_PREV) = -0.125*T*T*T * curr_R_a_hat;
    B_.block<3,3>(INDEX_THETA, INDEX_M_GYR_PREV) =
0.5*Eigen::Matrix3d::Identity()*T;
    B_.block<3,3>(INDEX_BETA, INDEX_M_GYR_PREV) = -0.25*T*T * curr_R_a_hat;
    // G13 & G33:
    B_.block<3,3>(INDEX_ALPHA, INDEX_M_ACC_CURR) = 0.25 * curr_R * T*T;
    B_.block<3,3>(INDEX_BETA, INDEX_M_ACC_CURR) = 0.5 * curr_R * T;
    // G14 & G24 & G34:
    B_.block<3,3>(INDEX_ALPHA, INDEX_M_GYR_CURR) = -0.125*T*T*T * curr_R_a_hat;
    B_.block<3,3>(INDEX_THETA, INDEX_M_GYR_CURR) =
0.5*Eigen::Matrix3d::Identity()*T;
    B_.block<3,3>(INDEX_BETA, INDEX_M_GYR_CURR) = -0.25*T*T * curr_R_a_hat;
    // G45
    B_.block<3,3>(INDEX_B_A, INDEX_R_ACC_PREV) = Eigen::Matrix3d::Identity() * T;
    // G56
    B_.block<3,3>(INDEX_B_G, INDEX_R_GYR_PREV) = Eigen::Matrix3d::Identity() * T;
    // TODO: 4. update P_:
    P_ = F_ * P_ * F_.transpose() + B_ * Q_ * B_.transpose();
    //
    // TODO: 5. update Jacobian:
```

```
    //
    J_ = F_ * J_;
}
```

oplusImpl：g2o更新状态量用的函数，除了姿态比较特殊外，其他直接用加法就行了。也要更新Vertex中对应的bias

```
virtual void oplusImpl(const double *update) override {
        //
        // TODO: do update
        //
        Eigen::Map<const Eigen::Matrix<double, 15, 1>> update_vec(update, 15);
        _estimate.pos += update_vec.block<3,1>(PRVAG::INDEX_POS, 0);
        _estimate.ori = _estimate.ori * Sophus::SO3d::exp(update_vec.block<3,1>
(PRVAG::INDEX_ORI, 0));
        _estimate.vel += update_vec.block<3,1>(PRVAG::INDEX_VEL, 0);
        _estimate.b_a += update_vec.block<3,1>(PRVAG::INDEX_B_A, 0);
        _estimate.b_g += update_vec.block<3,1>(PRVAG::INDEX_B_G, 0);
        updateDeltaBiases(update_vec.block<3,1>(PRVAG::INDEX_B_A, 0),
update_vec.block<3,1>(PRVAG::INDEX_B_G, 0));
    }
```

computeError：在Edge里面计算误差值，基本上就是先用bias更新预积分之后吧误差公式添上就行了。

```
virtual void computeError() override {
        g2o::VertexPRVAG* v0 = dynamic_cast<g2o::VertexPRVAG*>(_vertices[0]);
        g2o::VertexPRVAG* v1 = dynamic_cast<g2o::VertexPRVAG*>(_vertices[1]);

        const Eigen::Vector3d &pos_i = v0->estimate().pos;
        const Sophus::SO3d    &ori_i = v0->estimate().ori;
        const Eigen::Vector3d &vel_i = v0->estimate().vel;
        const Eigen::Vector3d &b_a_i = v0->estimate().b_a;
        const Eigen::Vector3d &b_g_i = v0->estimate().b_g;

        const Eigen::Vector3d &pos_j = v1->estimate().pos;
        const Sophus::SO3d    &ori_j = v1->estimate().ori;
        const Eigen::Vector3d &vel_j = v1->estimate().vel;
        const Eigen::Vector3d &b_a_j = v1->estimate().b_a;
        const Eigen::Vector3d &b_g_j = v1->estimate().b_g;

        //
        // TODO: update pre-integration measurement caused by bias change:
        //

        if (v0 -> isUpdated() ) {
            Eigen::Vector3d  d_b_a_i , d_b_g_i ;
            v0->getDeltaBiases(d_b_a_i, d_b_g_i);
            updateMeasurement(d_b_a_i,d_b_g_i);
        }

        //
        // TODO: compute error:
        //
```

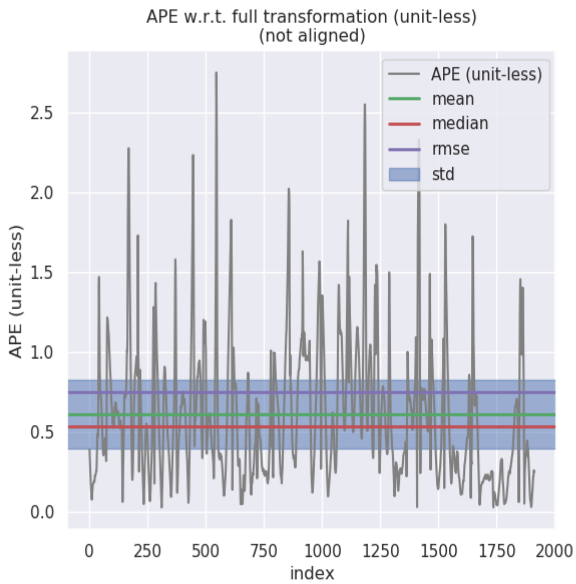
```
        const Eigen::Vector3d &alpha_ij = _measurement.block<3, 1>(INDEX_P, 0);
        const Eigen::Vector3d &theta_ij = _measurement.block<3, 1>(INDEX_R, 0);
        const Eigen::Vector3d &beta_ij = _measurement.block<3, 1>(INDEX_V, 0);
        // const Sophus::SO3d residual_ori_quat =
(Sophus::SO3d::exp(theta_ij).inverse() * ori_i.inverse() * ori_j);
        // const Eigen::Vector3d
residual_ori(residual_ori_quat.unit_quaternion().x(),
residual_ori_quat.unit_quaternion().y(),
residual_ori_quat.unit_quaternion().z());
        _error.block<3, 1>(INDEX_P, 0) = ori_i.matrix() * (pos_j - pos_i -
vel_i*T_ + 0.5*g_*T_*T_) - alpha_ij;
        _error.block<3, 1>(INDEX_R, 0) = (Sophus::SO3d::exp(theta_ij).inverse() *
ori_i.inverse() * ori_j).log();
        _error.block<3, 1>(INDEX_V, 0) = ori_i.inverse() * (vel_j - vel_i + g_ *
T_) - beta_ij;
        _error.block<3, 1>(INDEX_A, 0) = b_a_j - b_a_i;
        _error.block<3, 1>(INDEX_G, 0) = b_g_j - b_g_i;
    }
```

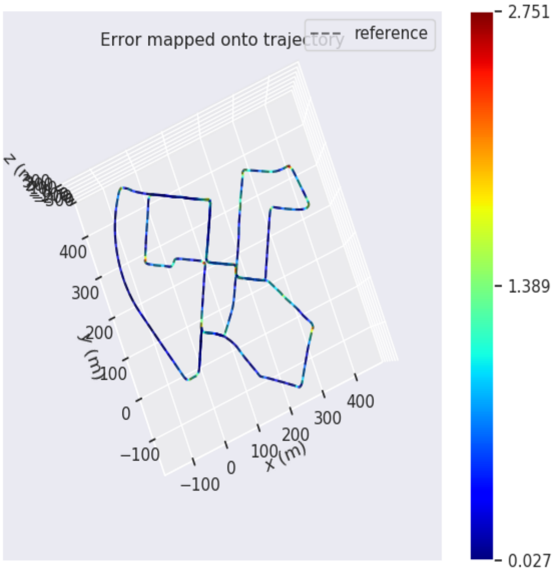
这一章有一点要注意就是pose*和T_nb的区别。pose*是当前的状态量(x)，T_nb是观测量(y)，所以在计算的时候基本都要用pose_，只有在计算观测量与状态量之差才会用T_nb。

效果：

用IMU预积分

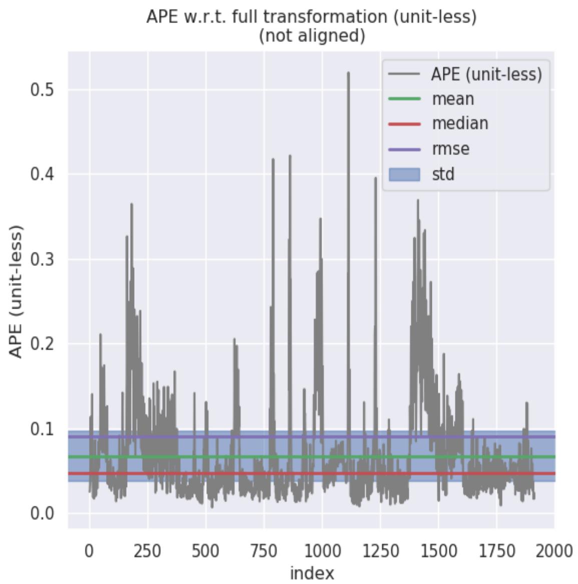
max	2.750943
mean	0.612079
median	0.531566
min	0.026736
rmse	0.747498
sse	1068.893565
std	0.429082

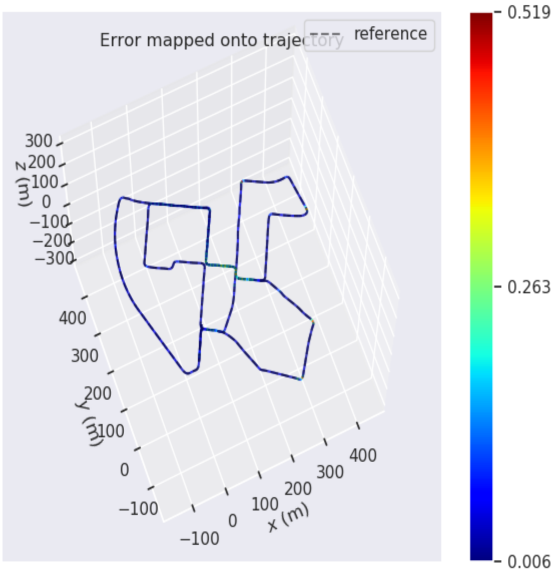




不用IMU预积分

max	0.519413
mean	0.067246
median	0.046663
min	0.006491
rmse	0.089805
sse	15.428082
std	0.059522





可以看出用IMU预积分反而效果变差了，应该是KITTI数据集的缘故

优秀部分：

编码器:

$$\begin{bmatrix} r_p \\ r_q \\ r_{bg} \end{bmatrix} = \begin{bmatrix} q^*_{wb_i}(p_{wb_j} - p_{wb_i}) - \alpha_{b_ib_j} \\ 2\left[q^*_{b_ib_j} \otimes (q^*_{wb_i} \otimes q_{wb_j})\right]_{xyz} \\ b^g_j - b^g_i \end{bmatrix}$$

由于文献以 α, θ, b_g 为状态量而非 p, q, b_g,
我们需推导 α, θ, b_g 的状态更新公式,以及 $\delta\alpha, \delta\theta, \delta b_g$.

① $\alpha_{b_ib_{k+1}} = \alpha_{b_ib_k} + q_{b_ib_k}\phi_k = \alpha_{b_ib_k} + R_{b_ib_k}\phi_k$

② $\tilde{\alpha}_{b_ib_{k+1}} = \tilde{\alpha}_{b_ib_k} + \tilde{q}_{b_ib_k}\tilde{\phi}_k$, $\phi_k = \frac{1}{2}(q_{b_ib_k}\phi^{b_k}_t + q_{b_ib_{k+1}}\phi^{b_{k+1}})$

③、$\tilde{\alpha}_{b_ib_{k+1}} = \alpha_{b_ib_{k+1}} + \delta\alpha_{b_ib_{k+1}}$

$\tilde{q}_{b_ib_k} = q_{b_ib_k} \otimes \delta q_{b_ib_k} = R_{b_ib_k}\exp(\delta\theta_k^\wedge)$

$\tilde{\phi}_k = \phi_k + \delta\phi_k$.

④ $\alpha_{b_{k+1}} + \delta\alpha_{b_{k+1}} = \alpha_{b_k} + \delta\alpha_{b_k} + R_{b_k}\exp(\delta\theta_k^\wedge)(\phi_k + \delta\phi_k)$

$\exp(\delta\theta_k^\wedge) \Rightarrow (I + \delta\theta_k^\wedge)$

⑤、 $\delta\alpha_{b_{k+1}} = \delta\alpha_{b_k} + R_{b_k}(\delta\phi_k + \delta\theta_k^\wedge\phi_k)$.

① $q_{b_{k+1}} = q_{b_k} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}w^b\delta t \end{bmatrix}$, $w^b = \frac{1}{2}\left[(w^{b} - b^g_k) + (w^{k+1} - b^g_{k+1})\right]$

② $\tilde{q}_{b_{k+1}} = \tilde{q}_{b_k} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}\tilde{w}^b\delta t \end{bmatrix}$

③、$\tilde{q}_{b_{k+1}} = q_{b_{k+1}} \otimes \delta q_{k+1}$

$\tilde{w}^b = \tilde{w}^b + \delta w^b$

④ $q_{b_{k+1}} \otimes \delta q_{k+1} = q_{b_k} \otimes \delta q_k \otimes \begin{bmatrix} 0 \\ \frac{1}{2}(w^b + \delta w^b)\delta t \end{bmatrix}$

⑤. $q_{b_k} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}w^b\delta t \end{bmatrix} \otimes \delta q_{k+1} = q_{b_k} \otimes \delta q_k \otimes \begin{bmatrix} 0 \\ \frac{1}{2}(w^b + \delta w^b)\delta t \end{bmatrix}$.

$\begin{bmatrix} 0 \\ \frac{1}{2}w^b\delta t \end{bmatrix} \otimes \delta q_{k+1} = \delta q_k \otimes \begin{bmatrix} 0 \\ \frac{1}{2}(w^b + \delta w^b)\delta t \end{bmatrix}$

$$\delta \xi_{k+1} = \begin{bmatrix} 0, & -\frac{1}{2}w^b\delta t^T \\ \frac{1}{2}w^b\delta t, & (\frac{1}{2}w^b\delta t)^\wedge \end{bmatrix}^{-1} \begin{bmatrix} 0, & -(\frac{1}{2}(w^b+\delta w^b)\delta t)^T \\ \frac{1}{2}(w^b+\delta w^b)\delta t & (\frac{1}{2}(w^b+\delta w^b)\delta t)^\wedge \end{bmatrix} \delta \xi_k$$

① $\theta_{k+1} = (\bar{w}_k - b_g)^\wedge \theta_k \delta t + \theta_k, \quad \bar{w}_k = \frac{1}{2}(w_k + w_{k+1})$

② $\tilde{\theta}_{k+1} = (\tilde{\bar{w}}_k - \tilde{b}_g)^\wedge \tilde{\theta}_k \delta t + \delta \tilde{\theta}_k$

③. $\tilde{\theta}_k = \theta_k + \delta\theta_k$

$\tilde{\bar{w}}_k = \bar{w}_k + \delta \bar{w}_k, \quad \overline{\delta w}_k = \frac{1}{2}(\delta w_k + \delta w_{k+1})$

$\tilde{b}_g = b_g + \delta b_g$

④. $\theta_{k+1} + \delta\theta_{k+1} = (\bar{w}_k + \overline{\delta w}_k + b_g + \delta b_g)(\theta_k + \delta\theta_k)^\wedge \delta t$
$+ \theta_k + \delta\theta_k$

⑤. $\delta\theta_{k+1} = (I - \bar{w}^\wedge \delta t)\delta\theta_k + \delta t \delta b_g + \overline{\delta w}_k$

① $b^g_{j+1} = b^g_j + \delta b_{jj+1} \delta t$

② $\tilde{b}^g_{j+1} = \tilde{b}^g_j$

③. $\tilde{b}^g_{j+1} = b^g_{j+1} + \delta b^g_{j+1} + \delta b_{jj+1}\delta t$

④. $b^g_{j+1} + \delta b^g_{j+1} = b^g_j + \delta b^g_j + \delta b_{jj+1}\delta t$

⑤ $\delta b^g_{j+1} = \delta b^g_j + \delta b_{jj+1}\delta t$

[illegible]

State Update Equation: $x = \begin{bmatrix} \delta\alpha_k \\ \delta\theta_k \\ \delta b_{gk} \end{bmatrix}$, $u = \begin{bmatrix} \delta\phi_k \\ \delta_{wk} \\ \delta_{wk+1} \\ \delta b^g_{bk} \end{bmatrix}$

$$\delta\alpha_{bk+1} = \delta\alpha_{bk} + R_{bk}(\delta\phi_k + \delta\theta_k^\wedge \phi_k).$$

$$\frac{\partial\alpha_{bk+1}}{\partial\alpha_{bk}} = I, \quad \frac{\partial\alpha_{bk+1}}{\partial\theta_k^\wedge} = -R_{bk}\phi_k^\wedge, \quad \frac{\partial\alpha_{bk+1}}{\partial b_g} = 0.$$

$$\frac{\partial\alpha_{bk+1}}{\partial\phi_k} = R_{bk}, \quad \frac{\partial\alpha_{bk+1}}{\partial w_k} = \frac{\partial\alpha_{bk+1}}{\partial w_{k+1}} = \frac{\partial\alpha_{bk+1}}{\partial b^g_k} = 0.$$

$$\delta\theta_{k+1} = (I - \bar{w}^\wedge\delta t)\delta\theta_k + \delta t\,\delta b_g + \delta \bar{w}_k$$

$$\frac{\partial\theta_{k+1}}{\partial\alpha_k} = 0, \quad \frac{\partial\theta_{k+1}}{\partial\theta_k} = I - \bar{w}^\wedge\delta t, \quad \frac{\partial\theta_{k+1}}{\partial b_g} = \delta t.$$

$$\frac{\partial\theta_{k+1}}{\partial\phi} = 0, \quad \frac{\partial\theta_{k+1}}{\partial w_k} = \frac{1}{2}\delta t\, I = \frac{\partial\theta_{k+1}}{\partial w_{k+1}}, \quad \frac{\partial\theta_{k+1}}{\partial b^g_k} = 0$$

$$\delta b^g_{j+1} = \delta b^g_j + \delta b_{j+1}\delta t$$

$$\frac{\partial b^g_{j+1}}{\partial\alpha_k} = 0 = \frac{\partial b^g_{j+1}}{\partial\theta_k}, \quad \frac{\partial b^g_{j+1}}{\partial b^g_k} = I.$$

$$\frac{\partial b^g_{j+1}}{\partial\phi} = 0 = \frac{\partial b_{j+1}}{\partial w_k} = \frac{\partial b^g_{j+1}}{\partial w_{k+1}}, \quad \frac{\partial b^g_{k+1}}{\partial \delta b^g_k} = \delta t I$$

$$\therefore F_k = \begin{bmatrix} I & -R_k[\phi_k]^\wedge & 0 \\ 0 & I - [\bar{w}]^\wedge\delta t & -I\delta t \\ 0 & 0 & I \end{bmatrix}$$

$$G_k = \begin{bmatrix} R_k & 0 & 0 & 0 \\ 0 & \frac{1}{2}\delta t I & \frac{1}{2}\delta t I & 0 \\ 0 & 0 & 0 & \delta t I \end{bmatrix}$$

Jacobian:

$$\frac{\partial r_p}{\partial p_i} = -R_{b_i}^{-1}, \quad \frac{\partial r_p}{\partial q_i} = R_{b_i}^{-1}(P_{w_j} - P_{w_i})$$

$$\frac{\partial r_p}{\partial b_i^g} = \frac{\partial \alpha_{b_i b_j}}{\partial \delta b_i^g} = -J^{\alpha}_{b_i^g}$$

$$\frac{\partial r_q}{\partial p_i} = 0, \quad \frac{\partial r_q}{\partial q_i} = -2\begin{bmatrix} 0 & I \end{bmatrix}\left[q_{b_j}^* \otimes q_{b_i}\right]_L \left[q_{b_i b_j}\right]_R \begin{bmatrix} 0 \\ \frac{1}{2}I \end{bmatrix}$$

$$\frac{\partial r_q}{\partial b_i^g} = -2\begin{bmatrix} 0 & I \end{bmatrix}\left[q_{b_j}^* \otimes q_{b_i} \otimes q_{b_i b_j}\right]_L \begin{bmatrix} 0 \\ \frac{1}{2}J^q_{b_i^g} \end{bmatrix}.$$

$$\frac{\partial r_{b_g^i}}{\partial p_i} = 0, \quad \frac{\partial r_{b_g^i}}{\partial q_i} = 0, \quad \frac{\partial r_{b_g^i}}{\partial b_g^i} = -I$$

$$\frac{\partial r_{p_i}}{\partial p_j} = R_{b_i}^{-1}, \quad \frac{\partial r_{p_i}}{\partial q_j} = 0, \quad \frac{\partial r_{p_i}}{\partial b_g^j} = 0$$

$$\frac{\partial r_{q_i}}{\partial p_j} = 0, \quad \frac{\partial r_{q_i}}{\partial q_j} = 2\begin{bmatrix} 0 & I \end{bmatrix}\left[q_{b_i b_j}^* \otimes q_{b_i}^* \otimes q_{b_i}\right]_L \begin{bmatrix} 0 \\ \frac{1}{2}I \end{bmatrix}$$

$$\frac{\partial r_{b_i^g}}{\partial p_j} = 0, \quad \frac{\partial r_{b_i^g}}{\partial q_j} = 0, \quad \frac{\partial r_{b_i^g}}{\partial b_j^g} = I.$$