第九次作业思路分享

主讲人 张松鹏

纲要

- IMU预积分公式推导
- IMU预积分代码
- 编码器预积分公式推导

IMU预积分公式推导

深蓝学院 shenlanxueyuan.com

- 残差

$$\begin{bmatrix} \mathbf{r}_p \\ \mathbf{r}_q \\ \mathbf{r}_v \\ \mathbf{r}_{ba} \\ \mathbf{r}_{bg} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{wb_i}^* \left(\mathbf{p}_{wb_j} - \mathbf{p}_{wb_i} - \mathbf{v}_i^w \Delta t + \frac{1}{2}\mathbf{g}^w \Delta t^2\right) - \boldsymbol{\alpha}_{b_i b_j} \\ 2\left[\mathbf{q}_{b_i b_j}^* \otimes \left(\mathbf{q}_{wb_i}^* \otimes \mathbf{q}_{wb_j}\right)\right]_{xyz} \\ \mathbf{q}_{wb_i}^* \left(\mathbf{v}_j^w - \mathbf{v}_i^w + \mathbf{g}^w \Delta t\right) - \boldsymbol{\beta}_{b_i b_j} \\ \mathbf{b}_j^a - \mathbf{b}_i^a \\ \mathbf{b}_j^g - \mathbf{b}_i^g \end{bmatrix}$$

- 优化变量

$$\left[\delta \mathbf{p}_{wb_i} \quad \delta \theta_{wb_i} \quad \delta \mathbf{v}_i^w \quad \delta \mathbf{b}_i^a \quad \delta \mathbf{b}_i^g\right]$$

$$\left[\delta \mathbf{p}_{wb_j} \quad \delta \theta_{wb_j} \quad \delta \mathbf{v}_j^w \quad \delta \mathbf{b}_j^a \quad \delta \mathbf{b}_j^g\right]$$

IMU预积分公式推导

●位置残差雅可比

$$
\begin{aligned}
\frac{\partial \mathbf{r}_p}{\partial \delta \mathbf{p}_{wb_i}} &= \frac{\partial - \mathbf{q}^*_{wb_i}(\mathbf{p}_{wb_i} + \delta \mathbf{p}_{wb_i})}{\partial \delta \mathbf{p}_{wb_i}} \\
&= -\mathbf{R}_{b_i w} \\
\frac{\partial \mathbf{r}_p}{\partial \delta \boldsymbol{\theta}_{wb_i}} &= \frac{\partial (\mathbf{q}_{wb_i} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}\delta \boldsymbol{\theta}_{wb_i} \end{bmatrix})^* (\mathbf{p}_{wb_j} - \mathbf{p}_{wb_i} - \mathbf{v}_i^w \Delta t + \frac{1}{2}\mathbf{g}^w \Delta t^2)}{\partial \delta \boldsymbol{\theta}_{wb_i}} \\
&= \frac{\partial (\mathbf{R}_{wb_i} \exp(\delta \boldsymbol{\theta}^\wedge_{wb_i}))^{-1} (\mathbf{p}_{wb_j} - \mathbf{p}_{wb_i} - \mathbf{v}_i^w \Delta t + \frac{1}{2}\mathbf{g}^w \Delta t^2)}{\partial \delta \boldsymbol{\theta}_{wb_i}} \\
&= \frac{\partial \exp((-\delta \boldsymbol{\theta}_{wb_i})^\wedge) \mathbf{R}_{b_i w} (\mathbf{p}_{wb_j} - \mathbf{p}_{wb_i} - \mathbf{v}_i^w \Delta t + \frac{1}{2}\mathbf{g}^w \Delta t^2)}{\partial \delta \boldsymbol{\theta}_{wb_i}} \\
&\approx \frac{\partial (\mathbf{I} - \delta \boldsymbol{\theta}^\wedge_{wb_i}) \mathbf{R}_{b_i w} (\mathbf{p}_{wb_j} - \mathbf{p}_{wb_i} - \mathbf{v}_i^w \Delta t + \frac{1}{2}\mathbf{g}^w \Delta t^2)}{\partial \delta \boldsymbol{\theta}_{wb_i}} \\
&= (\mathbf{R}_{b_i w} (\mathbf{p}_{wb_j} - \mathbf{p}_{wb_i} - \mathbf{v}_i^w \Delta t + \frac{1}{2}\mathbf{g}^w \Delta t^2))^\wedge \\
\frac{\partial \mathbf{r}_p}{\partial \delta \mathbf{v}_i^w} &= -\mathbf{R}_{b_i w} \Delta t \\
\frac{\partial \mathbf{r}_p}{\partial \delta \mathbf{b}_i^a} &= \frac{\partial - (\bar{\boldsymbol{\alpha}}_{b_i b_j} + \mathbf{J}^\alpha_{b_i^a} \delta \mathbf{b}_i^a + \mathbf{J}^\alpha_{b_i^g} \delta \mathbf{b}_i^g)}{\partial \delta \mathbf{b}_i^a} \\
&= -\mathbf{J}^\alpha_{b_i^a} \\
\frac{\partial \mathbf{r}_p}{\partial \delta \mathbf{b}_i^g} &= -\mathbf{J}^\alpha_{b_i^g}
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial \mathbf{r}_p}{\partial \delta \mathbf{b}_i^g} &= -\mathbf{J}^\alpha_{b_i^g} \\
\frac{\partial \mathbf{r}_p}{\partial \delta \mathbf{p}_{wb_j}} &= \mathbf{R}_{b_i w} \\
\frac{\partial \mathbf{r}_p}{\partial \delta \boldsymbol{\theta}_{wb_j}} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_p}{\partial \delta \mathbf{v}_j^w} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_p}{\partial \delta \mathbf{b}_j^a} &= \mathbf{0} \\
\frac{\partial \mathbf{r}_p}{\partial \delta \mathbf{b}_j^g} &= \mathbf{0}
\end{aligned}
$$

IMU预积分公式推导

- 姿态残差雅可比

$$
\frac{\partial \mathbf{r}_\theta}{\partial \delta \mathbf{p}_{wb_i}} = \mathbf{0}
$$

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

$$
\frac{\partial \mathbf{r}_\theta}{\partial \delta \mathbf{v}_i^w} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_\theta}{\partial \delta \mathbf{b}_i^a} = \mathbf{0}
$$

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

$$
\frac{\partial \mathbf{r}_\theta}{\partial \delta \mathbf{p}_{wb_j}} = \mathbf{0}
$$

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

$$
\frac{\partial \mathbf{r}_\theta}{\partial \delta \mathbf{v}_j^w} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_\theta}{\partial \delta \mathbf{b}_j^a} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_\theta}{\partial \delta \mathbf{b}_j^g} = \mathbf{0}
$$

IMU预积分公式推导

- 速度残差雅可比

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{p}_{wb_i}} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_v}{\partial \delta \theta_{wb_i}} = (\mathbf{R}_{b_i w}(\mathbf{v}_{wb_j} - \mathbf{v}_{wb_i} + \mathbf{g}^w \Delta t))^\wedge$$

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

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{b}_i^a} = \frac{\partial - (\bar{\beta}_{b_i b_j} + \mathbf{J}^\beta_{b_i^a} \delta \mathbf{b}_i^a + \mathbf{J}^\beta_{b_i^g} \delta \mathbf{b}_i^g)}{\partial \delta \mathbf{b}_i^a} = -\mathbf{J}^\beta_{b_i^a}$$

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{b}_i^g} = -\mathbf{J}^\beta_{b_i^g}$$

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{p}_{wb_j}} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_v}{\partial \delta \theta_{wb_j}} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{v}_j^w} = \mathbf{R}_{b_i w}$$

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{b}_j^a} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_v}{\partial \delta \mathbf{b}_j^g} = \mathbf{0}$$

IMU预积分公式推导

- 加速度零偏残差雅可比

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \mathbf{p}_{wb_i}} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \theta_{wb_i}} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \mathbf{v}_i^w} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \mathbf{b}_i^a} = \frac{\partial(\mathbf{b}_j^a - (\mathbf{b}_i^a + \delta \mathbf{b}_i^a))}{\partial \delta \mathbf{b}_i^a} = -\mathbf{I}$$

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \mathbf{b}_i^g} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \mathbf{p}_{wb_j}} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \theta_{wb_j}} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \mathbf{v}_j^w} = \mathbf{0}$$

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \mathbf{b}_j^a} = \mathbf{I}$$

$$\frac{\partial \mathbf{r}_{b^a}}{\partial \delta \mathbf{b}_j^g} = \mathbf{0}$$

IMU预积分公式推导

深蓝学院
shenlanxueyuan.com

- 角速度零偏残差雅可比

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \mathbf{p}_{wb_i}} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \theta_{wb_i}} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \mathbf{v}_i^w} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \mathbf{b}_i^a} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \mathbf{b}_i^g} = \frac{\partial(\mathbf{b}_j^g - (\mathbf{b}_i^g + \delta \mathbf{b}_i^g))}{\partial \delta \mathbf{b}_i^g} = -\mathbf{I}
$$

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \mathbf{p}_{wb_j}} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \theta_{wb_j}} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \mathbf{v}_j^w} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \mathbf{b}_j^a} = \mathbf{0}
$$

$$
\frac{\partial \mathbf{r}_{b^g}}{\partial \delta \mathbf{b}_j^g} = \mathbf{I}
$$

IMU预积分代码

- 任务

填写以下3个文件中的TODO部分：

lidar_localization/src/models/pre_integrator/imu_pre_integrator.cpp

lidar_localization/include/lidar_localization/models/graph_optimizer/g2o/edge/edge_prvag_imu_pre_integration.hpp

lidar_localization/include/lidar_localization/models/graph_optimizer/g2o/vertex/vertex_prvag.hpp

IMU预积分代码

- imu_pre_integrator.cpp

名义值更新：中值积分

```
//
// 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);
state.theta_ij_ = state.theta_ij_ * d_theta_ij;
curr_theta_ij = state.theta_ij_;
// 3. get a_mid:
a_mid = 0.5 * (prev_theta_ij * prev_a + curr_theta_ij * 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;
```

IMU预积分代码

- imu_pre_integrator.cpp

误差值更新：中间值

```
//
// 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);
```

IMU预积分代码

- imu_pre_integrator.cpp

误差值更新：F矩阵

```
// TODO: 2. set up F:
//
// F12 & F32:
F_.block<3, 3>(INDEX_ALPHA, INDEX_THETA) = -0.25 * T * (prev_R_a_hat + curr_R_a_hat * dR_inv);
F_.block<3, 3>(INDEX_BETA, INDEX_THETA) = -0.5 * (prev_R_a_hat + curr_R_a_hat * dR_inv);
// F14 & F34:
F_.block<3, 3>(INDEX_ALPHA, INDEX_B_A) = -0.25 * T *(prev_R + curr_R);
F_.block<3, 3>(INDEX_BETA, INDEX_B_A) = -0.5 * (prev_R + curr_R);
// F15 & F35:
F_.block<3, 3>(INDEX_ALPHA, INDEX_B_G) = 0.25 * T * T * curr_R_a_hat;
F_.block<3, 3>(INDEX_BETA, INDEX_B_G) = 0.5 * T * curr_R_a_hat;
// F22:
F_.block<3, 3>(INDEX_THETA, INDEX_THETA) = -Sophus::SO3d::hat(w_mid);
```

IMU预积分代码

- imu_pre_integrator.cpp

误差值更新：B矩阵

```
// TODO: 3. set up G:
//
// G11 & G31:
B_.block<3, 3>(INDEX_ALPHA, INDEX_M_ACC_PREV) = 0.25 * T * prev_R;
B_.block<3, 3>(INDEX_BETA, INDEX_M_ACC_PREV) = 0.5 * prev_R;
// G12 & G32:
B_.block<3, 3>(INDEX_ALPHA, INDEX_M_GYR_PREV) = -0.125 * T * T * curr_R_a_hat;
B_.block<3, 3>(INDEX_BETA, INDEX_M_GYR_PREV) = -0.25 * T * curr_R_a_hat;
// G13 & G33:
B_.block<3, 3>(INDEX_ALPHA, INDEX_M_ACC_CURR) = 0.25 * T * curr_R;
B_.block<3, 3>(INDEX_BETA, INDEX_M_ACC_CURR) = 0.5 * curr_R;
// G14 & G34:
B_.block<3, 3>(INDEX_ALPHA, INDEX_M_GYR_CURR) = -0.125 * T * T * curr_R_a_hat;
B_.block<3, 3>(INDEX_BETA, INDEX_M_GYR_CURR) = -0.25 * T * curr_R_a_hat;
```

IMU预积分代码

- imu_pre_integrator.cpp

误差值更新：P和J矩阵

```
// TODO: 4. update P_:
MatrixF F = MatrixF::Identity() + T * F_;
MatrixB B = T * B_;
P_ = F * P_ * F.transpose() + B * Q_ * B.transpose();
//
// TODO: 5. update Jacobian:
//
J_ = F * J_;
```

IMU预积分代码

- edge_prvag_imu_pre_integration.hpp

computeError函数

```
// 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);
_error.block<3, 1>(INDEX_P, 0) = ori_i.inverse() * (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;
```

IMU预积分代码

- vertex_prvag.hpp

oplusImpl函数

```
// TODO: do update
//
_estimate.pos += Eigen::Vector3d(
    update[PRVAG::INDEX_POS + 0], update[PRVAG::INDEX_POS + 1], update[PRVAG::INDEX_POS + 2]
);
_estimate.ori = _estimate.ori * Sophus::SO3d::exp(
    Eigen::Vector3d(
        update[PRVAG::INDEX_ORI + 0], update[PRVAG::INDEX_ORI + 1], update[PRVAG::INDEX_ORI + 2]
    )
);
_estimate.vel += Eigen::Vector3d(
    update[PRVAG::INDEX_VEL + 0], update[PRVAG::INDEX_VEL + 1], update[PRVAG::INDEX_VEL + 2]
);

Eigen::Vector3d d_b_a_i(
    update[PRVAG::INDEX_B_A + 0], update[PRVAG::INDEX_B_A + 1], update[PRVAG::INDEX_B_A + 2]
);
Eigen::Vector3d d_b_g_i(
    update[PRVAG::INDEX_B_G + 0], update[PRVAG::INDEX_B_G + 1], update[PRVAG::INDEX_B_G + 2]
);

_estimate.b_a += d_b_a_i;
_estimate.b_g += d_b_g_i;

updateDeltaBiases(d_b_a_i, d_b_g_i);
```

编码器预积分公式推导

深蓝学院 shenlanxueyuan.com

- 残差

$$\begin{bmatrix} \mathbf{r}_p \\ \mathbf{r}_q \\ \mathbf{r}_{bg} \end{bmatrix} = \begin{bmatrix} \mathbf{q}^*_{wb_i}\left(\mathbf{p}_{wb_j} - \mathbf{p}_{wb_i}\right) - \boldsymbol{\alpha}_{b_ib_j} \\ 2\left[\mathbf{q}^*_{b_ib_j} \otimes \left(\mathbf{q}^*_{wb_i} \otimes \mathbf{q}_{wb_j}\right)\right]_{xyz} \\ \mathbf{b}^g_j - \mathbf{b}^g_i \end{bmatrix}$$

- 优化变量

$$\begin{bmatrix}\delta\mathbf{p}_{wb_i} & \delta\theta_{wb_i} & \delta\mathbf{b}^g_i\end{bmatrix}$$

$$\begin{bmatrix}\delta\mathbf{p}_{wb_j} & \delta\theta_{wb_j} & \delta\mathbf{b}^g_j\end{bmatrix}$$

- 推导过程请见附件。

在线问答

Q&A

感谢各位聆听！

Thanks for Listening