第八章作业讲解

主讲人 Teamo

纲要

- 第一部分：融合雷达位姿和编码器速度
- 第二部分：融合GPS位置与编码器速度

融合雷达位姿和编码器速度

$$\delta\bar{\boldsymbol{v}}_b = \tilde{\boldsymbol{v}}^b - \boldsymbol{v}^b = \tilde{\boldsymbol{R}}_{bw}\tilde{\boldsymbol{v}}^w - \begin{bmatrix} \boldsymbol{v}_m \\ 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{G}_t = \begin{bmatrix} \boldsymbol{I}_3 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{R}_{bw} & [\boldsymbol{v}^b]_\times & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_3 & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

$$\boldsymbol{C}_t = \begin{bmatrix} \boldsymbol{I}_3 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_3 \end{bmatrix}$$

融合雷达位姿和编码器速度

深蓝学院
shenlanxueyuan.com

```
case MeasurementType::POSE_VEL:
  //
  CorrectErrorEstimationPoseVel(measurement.T_nb,measurement.v_b,
                                measurement.w_b,Y,G,K);
  //
  break;
```

融合雷达位姿和编码器速度

深蓝学院
shenlanxueyuan.com

```cpp
void ErrorStateKalmanFilter::CorrectErrorEstimationPoseVel(
    const Eigen::Matrix4d &T_nb, const Eigen::Vector3d &v_b, const Eigen::Vector3d &w_b,
    Eigen::VectorXd &Y, Eigen::MatrixXd &G, Eigen::MatrixXd &K
) {
    // 设置测量值:
    Eigen::Vector3d P_nn_obs = pose_.block<3, 1>(0,3) - T_nb.block<3, 1>(0,3);
    Eigen::Matrix3d C_nn_obs = T_nb.block<3, 3>(0,0).transpose()* pose_.block<3, 3>(0,0);
    Eigen::Vector3d v_bb_obs = pose_.block<3, 3>(0,0).transpose()*vel_ - v_b;

    YPoseVel_.block<3, 1>(0, 0) = P_nn_obs;
    YPoseVel_.block<3, 1>(3, 0) = Sophus::SO3d::vee(C_nn_obs - Eigen::Matrix3d::Identity() );
    YPoseVel_.block<3, 1>(6, 0) = v_bb_obs;

    Y = YPoseVel_;

    // 设置观测矩阵:
    GPoseVel_.block<3, 3>(6, kIndexErrorVel) =  pose_.block<3, 3>(0,0).transpose();
    GPoseVel_.block<3, 3>(6, kIndexErrorOri) = Sophus::SO3d::hat(pose_.block<3, 3>(0,0).transpose()*vel_);

    G = GPoseVel_;

    // 计算卡尔曼增益:
    MatrixRPoseVel R = GPoseVel_*P_*GPoseVel_.transpose() + RPoseVel_;
    K = P_*GPoseVel_.transpose()*R.inverse();
}
```

融合雷达位姿和编码器速度

```
// 计算后验误差估计和协方差矩阵
P_ = (MatrixP::Identity() - K*G)*P_;
X_ = X_+K*(Y-G*X_);
```

纲要

- 第一部分：融合雷达位姿和编码器速度
- 第二部分：融合GPS位置与编码器速度

融合GPS位置与编码器速度

深蓝学院
shenlanxueyuan.com

```
case MeasurementType::POSI_VEL:
  //
  CorrectErrorEstimationPosiVel(measurement.T_nb,measurement.v_b,
                                measurement.w_b,Y,G,K);
  //
  break;
```

```
void ErrorStateKalmanFilter::CorrectErrorEstimationPosiVel(
    const Eigen::Matrix4d &T_nb, const Eigen::Vector3d &v_b, const Eigen::Vector3d &w_b,
    Eigen::VectorXd &Y, Eigen::MatrixXd &G, Eigen::MatrixXd &K
) {
    // parse measurement:
    Eigen::Vector3d P_nn_obs = pose_.block<3, 1>(0,3) - T_nb.block<3, 1>(0,3);
    Eigen::Vector3d v_bb_obs = pose_.block<3, 3>(0,0).transpose()*vel_ - v_b;
    YPosiVel_.block<3, 1>(0, 0) = P_nn_obs;
    YPosiVel_.block<3, 1>(3, 0) = v_bb_obs;
    Y = YPosiVel_;

    // set measurement equation:
    GPosiVel_.block<3, 3>(3, kIndexErrorVel) =  pose_.block<3, 3>(0,0).transpose();
    GPosiVel_.block<3, 3>(3, kIndexErrorOri) =  Sophus::SO3d::hat(pose_.block<3, 3>(0,0).transpose()*vel_); ;
    G = GPosiVel_;
    // set Kalman gain:
    MatrixRPosiVel R = GPosiVel_*P_*GPosiVel_.transpose() + RPosiVel_;
    K = P_*GPosiVel_.transpose()*R.inverse();
}
```

融合GPS位置与编码器速度

深蓝学院 shenlanxueyuan.com

```python
'gps': {
    'no_error': {
        'stdp': np.array([0.0, 0.0, 0.0]),
        'stdv': np.array([0.0, 0.0, 0.0])
    },
    'high_accuracy': {
        'stdp': np.array([0.10, 0.10, 0.10]),
        'stdv': np.array([0.01, 0.01, 0.01])
    },
    'mid_accuracy': {
        'stdp': np.array([0.50, 0.50, 0.50]),
        'stdv': np.array([0.02, 0.02, 0.02])
    },
    'low_accuracy': {
        'stdp': np.array([1.00, 1.00, 1.00]),
        'stdv': np.array([0.05, 0.05, 0.05])
    }
},
```

```python
'odo': {
    'no_error': {
        'scale': 1.00,
        'stdv': 0.0
    },
    'high_accuracy': {
        'scale': 1.00,
        'stdv': 0.01
    },
    'mid_accuracy': {
        'scale': 1.00,
        'stdv': 0.05
    },
    'low_accuracy': {
        'scale': 1.00,
        'stdv': 0.10
    }
}
```

```yaml
measurement:
    pose:
        pos: 1.0e-4
        ori: 1.0e-4
    pos: 1.0e-4
    vel: 2.5e-3
```

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening