第5章作业

一. 雅可比推导

安装误差 T（下三角形式），刻度系数误差K，零偏B

$$T=\begin{bmatrix}1 & 0 & 0\\ s_{xz} & 1 & 0\\ -s_{xy} & s_{yx} & 1\end{bmatrix}\quad=\begin{bmatrix}1 & 0 & 0\\ s_1 & 1 & 0\\ -s_2 & s_3 & 1\end{bmatrix}$$

$$K_a'=\begin{bmatrix}K_{ax}' & & \\ & K_{ay}' & \\ & & K_{az}'\end{bmatrix}=K_a^{-1}=\begin{bmatrix}\frac{1}{K_{ax}} & & \\ & \frac{1}{K_{ev}} & \\ & & \frac{1}{K_{az}}\end{bmatrix}=\begin{bmatrix}\mathrm{k}_1 & 0 & 0\\ 0 & k_2 & 0\\ 0 & 0 & k_3\end{bmatrix}$$

$$T=\begin{bmatrix}1 & 0 & 0\\ s_1 & 1 & 0\\ -s_2 & s_3 & 1\end{bmatrix}\quad K=\begin{bmatrix}\mathrm{k}_1 & 0 & 0\\ 0 & k_2 & 0\\ 0 & 0 & k_3\end{bmatrix}\quad B=\begin{bmatrix}b_x\\ b_y\\ b_z\end{bmatrix}$$

待估计参数如下

$$\theta^{acc}=\begin{bmatrix}s_1 & s_2 & s_3 & k_1 & k_2 & k_3 & b_x & b_y & b_z\end{bmatrix}$$

给定加速度读数为 X，对应的真实值为 X‘，其计算公式如下:

$$X'=T*K*(X-B)$$

残差:

$$f\left(\theta^{acc}\right)=\|g\|_2-\|X'\|_2$$

雅可比按照链式求导分解:

$$\frac{\partial f}{\partial\theta^{acc}}=\frac{\partial f}{\partial\|X'\|_2}\frac{\partial\|X'\|_2}{\partial X'}\frac{\partial X'}{\partial\theta^{acc}}$$

$$X=\begin{bmatrix}A_x\\ A_y\\ A_z\end{bmatrix}\quad X'=T*K*(X-B)=\begin{bmatrix}k_1\left(A_x-b_x\right)\\ s_1k_1\left(A_x-b_x\right)+k_2\left(A_y-b_y\right)\\ -s_2k_1\left(A_x-b_x\right)+s_3k_2\left(A_y-b_y\right)+k_3\left(A_z-b_z\right)\end{bmatrix}$$

$$\frac{\partial f}{\partial\|X'\|_2}=\frac{\partial\left(\|g\|_2-\|X'\|_2\right)}{\partial\|X'\|_2}=-1$$

$$\frac{\partial\|X'\|_2}{\partial X'}=\frac{X'}{\|X'\|_2}$$

$$\frac{\partial X'}{\partial\theta^{acc}}=\begin{bmatrix}0 & 0 & 0 & A_x-b_x & 0 & 0 & -k_1 & 0 & 0\\ k_1\left(A_x-b_x\right) & 0 & 0 & s_1\left(A_x-b_x\right) & A_y-b_y & 0 & -s_1k_1 & -k_2 & 0\\ 0 & -k_1\left(A_x-b_x\right) & k_2\left(A_y-b_y\right) & -s_2\left(A_x-b_x\right) & s_3\left(A_y-b_y\right) & A_z-b_z & s_2k_1 & -s_3k_2 & -k_3\end{bmatrix}$$

二.标定实验

代码补全

FILE： IMU_TK/src/calibration.cpp

TODO1

加速度残差自动求导修改为下三角模型：

```
template <typename _T1> struct MultiPosAccResidual
{
  MultiPosAccResidual(
    const _T1 &g_mag,
    const Eigen::Matrix< _T1, 3 , 1> &sample
  ) : g_mag_(g_mag), sample_(sample){}

  template <typename _T2>
  bool operator() (
    const _T2* const params,
    _T2* residuals
  ) const {
    Eigen::Matrix<_T2, 3, 1> raw_samp(
      _T2(sample_(0)),
      _T2(sample_(1)),
      _T2(sample_(2))
    );

    /* Apply undistortion transform to accel measurements
         mis_mat_ <<  _T(1)   , -mis_yz  ,  mis_zy  ,
                       mis_xz ,  _T(1)   , -mis_zx  ,
                      -mis_xy ,  mis_yx  ,  _T(1)   ;

       scale_mat_ <<   s_x  ,   _T(0)  ,  _T(0) ,
                      _T(0) ,    s_y   ,  _T(0) ,
                      _T(0) ,   _T(0)  ,   s_z  ;

        bias_vec_ <<  b_x , b_y , b_z ;

        define:
           ms_mat_ = mis_mat_*scale_mat_

        then the undistortion transform:
                X' = T*K*(X - B)

        can be implemented as:

       unbias_data = ms_mat_*(raw_data - bias_vec_)

     * assume body frame same as accelerometer frame,
     * so bottom left params in the misalignment matris are set to zero */
    // CalibratedTriad_<_T2> calib_triad(
    //   //
    //   // TODO: implement lower triad model here
    //   //
    //   // mis_yz, mis_zy, mis_zx:
    //   params[0], params[1], params[2],
    //   // mis_xz, mis_xy, mis_yx:
    //   _T2(0), _T2(0), _T2(0),
  //   //    s_x,    s_y,    s_z:
  //   params[3], params[4], params[5],
  //   //    b_x,    b_y,    b_z:
  //   params[6], params[7], params[8]
  // );

    CalibratedTriad_<_T2> calib_triad(                     //    12 个内参
   //
   // TODO: implement lower triad model here         初始化内参
   //
   // mis_yz, mis_zy, mis_zx:                          安装误差
   _T2(0), _T2(0), _T2(0),                                //   因为使用的是没有
转台的模式，所以可以省去安装误差3个参数
   // mis_xz, mis_xy, mis_yx:
   params[0], params[1], params[2],
   //    s_x,    s_y,    s_z:
   params[3], params[4], params[5],          //   标度因素
   //    b_x,    b_y,    b_z:
   params[6], params[7], params[8]           //   零偏
  );
  // apply undistortion transform:
  Eigen::Matrix< _T2, 3 , 1> calib_samp = calib_triad.unbiasNormalize(
raw_samp );

  residuals[0] = _T2 (g_mag_) - calib_samp.norm();

  return true;
 }
```

TODO2

```
  // TODO: implement lower triad model here         实现下三角模型
  //
   //origin   安装误差   上三角模型
  // acc_calib_params[0] = init_acc_calib_.misYZ();
  // acc_calib_params[1] = init_acc_calib_.misZY();
  // acc_calib_params[2] = init_acc_calib_.misZX();

  /*new  按照课程的公式，推导，改为下三角模型*/
  acc_calib_params[0] = init_acc_calib_.misXZ();
  acc_calib_params[1] = init_acc_calib_.misXY();
  acc_calib_params[2] = init_acc_calib_.misYX();

  acc_calib_params[3] = init_acc_calib_.scaleX();                //     标度
因素
  acc_calib_params[4] = init_acc_calib_.scaleY();
  acc_calib_params[5] = init_acc_calib_.scaleZ();

  acc_calib_params[6] = init_acc_calib_.biasX();                 //   零偏
  acc_calib_params[7] = init_acc_calib_.biasY();
  acc_calib_params[8] = init_acc_calib_.biasZ();
```

TODO3

```
 acc_calib_ = CalibratedTriad_<_T>(
    //
    // TODO: implement lower triad model here
    //
    // min_cost_calib_params[0],
    // min_cost_calib_params[1],
    // min_cost_calib_params[2],
    // 0,0,0,
    // min_cost_calib_params[3],
    // min_cost_calib_params[4],
    // min_cost_calib_params[5],
    // min_cost_calib_params[6],
    // min_cost_calib_params[7],
    // min_cost_calib_params[8]
       0,0,0,

    min_cost_calib_params[0],
    min_cost_calib_params[1],
    min_cost_calib_params[2],
    min_cost_calib_params[3],
    min_cost_calib_params[4],
    min_cost_calib_params[5],
    min_cost_calib_params[6],
    min_cost_calib_params[7],
    min_cost_calib_params[8]
  );
```

解析式求导

```
template <typename _T1>
class  MultiPosAccResidual_Analytical   :  public   ceres::SizedCostFunction<1,
9>   {      //  优化参数维度 residual[0]：1    输入维度：9
public:
        const _T1 g_mag_;
        const Eigen::Matrix< _T1, 3 , 1> sample_;
        MultiPosAccResidual_Analytical( const  _T1 &g_mag,

                            const Eigen::Matrix<_T1,3,1>&sample)
                        : g_mag_(g_mag),     sample_(sample)   { }

virtual   bool   Evaluate(double  const  *const   *parameters,

               double   *residuals,

               double   **jacobians )  const                             //
  定义残差模型
{

        Eigen::Matrix<double, 3, 1> raw_samp(             //  观测数据
      double(sample_(0)),
      double(sample_(1)),
      double(sample_(2))
    );
```

```
         CalibratedTriad_<double> calib_triad(                       //    12 个内参
        //
        // TODO: implement lower triad model here          初始化内参
        //
        // mis_yz, mis_zy, mis_zx:                            安装误差
        double(0), double(0), double(0),                                  //  因为
使用的是没有转台的模式，所以可以省去安装误差3个参数
        // mis_xz, mis_xy, mis_yx:
        parameters[0][0], parameters[0][1], parameters[0][2],

        //    s_x,     s_y,     s_z:
        parameters[0][3], parameters[0][4], parameters[0][5],           //  标度因素

        //    b_x,     b_y,     b_z:
        parameters[0][6], parameters[0][7], parameters[0][8]            //  零偏
      );

        // apply undistortion transform:
      Eigen::Matrix< double, 3 , 1> calib_samp = calib_triad.unbiasNormalize(
raw_samp );

      residuals[0] = (double)g_mag_  -   calib_samp.norm();      //  残差

      if(jacobians  !=   nullptr)
      {
                    if (jacobians[0]  !=   nullptr)
                    {

                          /*计算雅克比*/
                                //   安装误差
                            double  S1 =     parameters[0][0];
                            double  S2 =     parameters[0][1];
                            double  S3 =     parameters[0][2];
                            //   标度因素
                            double  K1 =     parameters[0][3];
                            double  K2 =     parameters[0][4];
                            double  K3 =     parameters[0][5];
                            //    零偏
                            double  Bx =     parameters[0][6];
                            double  By =     parameters[0][7];
                            double  Bz =     parameters[0][8];

                            // 计算出的真值输出
                            double  Ax =  raw_samp[0];
                            double  Ay =  raw_samp[1];
                            double  Az =  raw_samp[2];

                             Eigen::Matrix< double, 1, 9> Jacobian ;
                            Eigen::Vector3d  samp_norm  =     calib_samp  /
(calib_samp.norm() )  ;     //  输入数据的单位向量
                            Eigen::Matrix< double, 3, 9> J_theta  =
 Eigen::Matrix<double,  3,   9>::Zero();

                            J_theta(0,3) =  (Ax - Bx);
                            J_theta(0,6)  =  -K1;

                            J_theta(1,0)  =  K1*(Ax - Bx);
                            J_theta(1,3)  =  S1*(Ax - Bx);
                          J_theta(1,4)  =  Ay - By;
                          J_theta(1,6)  =  -S1*K1;
                          J_theta(1,7)  =  -K2;

                          J_theta(2,1)  =  -K1*(Ax - Bx);
                          J_theta(2,2)  =  K2*(Ay - By);
                          J_theta(2,3)  =  -S2*(Ax - Bx);
                          J_theta(2,4)  =  S3*(Ay - By);
                          J_theta(2,5)  =  Az - Bz;
                          J_theta(2,6)  =  S2*K1;
                          J_theta(2,7)  =  -S3*K2;
                          J_theta(2,8)  =  -K3;
                          Jacobian    =   - samp_norm.transpose()  * J_theta  ;

                          jacobians[0][0] = Jacobian(0,0);
                          jacobians[0][1] = Jacobian(0,1);
                          jacobians[0][2] = Jacobian(0,2);
                          jacobians[0][3] = Jacobian(0,3);
                          jacobians[0][4] = Jacobian(0,4);
                          jacobians[0][5] = Jacobian(0,5);
                          jacobians[0][6] = Jacobian(0,6);
                          jacobians[0][7] = Jacobian(0,7);
                          jacobians[0][8] = Jacobian(0,8);
                  }
    }
    return  true;
}

} ;
```

调用

使用宏，若 #define autograde 则使用自动求导， 反之使用解析式求导

```
        #ifdef  autograde
          ceres::CostFunction* cost_function = MultiPosAccResidual<_T>::Create (
            g_mag_, static_samples[i].data()
          );
      #else
          ceres::CostFunction *cost_function = new
MultiPosAccResidual_Analytical<_T>(
            g_mag_, static_samples[i].data());
      #endif
```

运行

代码运行命令：

```
./test_imu_calib ./test_data/xsens_acc.mat  ./test_data/xsens_gyro.mat
```

效果

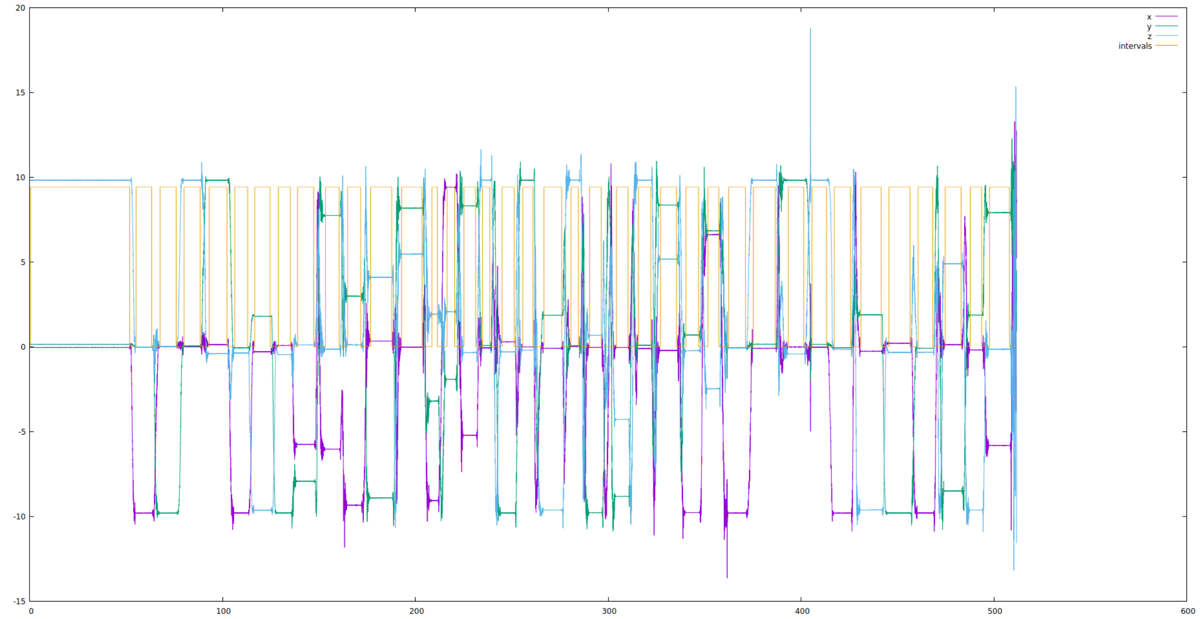
解析式求导

```
Importing IMU data from the Matlab matrix file : ./test_data/xsens_acc.mat
Importing IMU data from the Matlab matrix file : ./test_data/xsens_gyro.mat
Accelerometer Calibration: Calibrating...
Accelerometers Calibration: Extracted 42 intervals using threshold multiplier 2 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.399567e+10    0.00e+00    3.17e+10   0.00e+00   0.00e+00  1.00e+04
0    6.12e-04    8.60e-04
   1  2.700191e+03    3.40e+10    5.32e+06   2.63e+00   1.00e+00  3.00e+04
1    1.29e-03    2.17e-03
   2  2.278048e+01    2.68e+03    6.95e+05   7.16e+02   9.92e-01  9.00e+04
1    1.05e-03    3.23e-03
   3  1.366645e-01    2.26e+01    7.18e+03   3.18e+01   1.00e+00  2.70e+05
1    1.05e-03    4.29e-03
   4  1.347244e-01    1.94e-03    1.18e+00   4.74e-01   1.00e+00  8.10e+05
1    1.07e-03    5.36e-03
residual 0.134724
Accelerometers Calibration: Extracted 40 intervals using threshold multiplier 3 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.189931e+10    0.00e+00    2.76e+10   0.00e+00   0.00e+00  1.00e+04
0    5.72e-04    7.62e-04
   1  2.343289e+03    3.19e+10    3.46e+06   2.60e+00   1.00e+00  3.00e+04
1    1.01e-03    1.78e-03
   2  2.118105e+01    2.32e+03    6.06e+05   7.18e+02   9.91e-01  9.00e+04
1    1.01e-03    2.80e-03
   3  1.332983e-01    2.10e+01    6.87e+03   3.17e+01   1.00e+00  2.70e+05
1    1.01e-03    3.81e-03
   4  1.313451e-01    1.95e-03    1.04e+00   4.89e-01   1.00e+00  8.10e+05
1    1.01e-03    4.82e-03
residual 0.131345
Accelerometers Calibration: Extracted 39 intervals using threshold multiplier 4 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.112881e+10    0.00e+00    2.73e+10   0.00e+00   0.00e+00  1.00e+04
0    5.57e-04    7.03e-04
   1  2.310335e+03    3.11e+10    3.40e+06   2.61e+00   1.00e+00  3.00e+04
1    9.86e-04    1.70e-03
   2  2.117401e+01    2.29e+03    6.06e+05   7.18e+02   9.91e-01  9.00e+04
1    9.74e-04    2.68e-03
   3  1.279232e-01    2.10e+01    6.76e+03   3.00e+01   1.00e+00  2.70e+05
1    1.05e-03    3.73e-03
   4  1.261258e-01    1.80e-03    9.99e-01   4.65e-01   1.00e+00  8.10e+05
1    1.07e-03    4.81e-03
residual 0.126126
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 5 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
```

```
   0  3.020722e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    5.47e-04    7.57e-04
   1  2.236142e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    9.71e-04    1.74e-03
   2  2.057339e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    1.02e-03    2.77e-03
   3  1.223695e-01    2.05e+01    6.81e+03   3.01e+01   1.00e+00  2.70e+05
1    1.10e-03    3.87e-03
   4  1.205582e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    1.10e-03    4.98e-03
residual 0.120558
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 6 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020735e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    8.51e-04    1.04e-03
   1  2.236228e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    1.19e-03    2.27e-03
   2  2.057615e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    9.54e-04    3.23e-03
   3  1.219430e-01    2.05e+01    6.81e+03   3.01e+01   1.00e+00  2.70e+05
1    9.53e-04    4.19e-03
   4  1.201312e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    1.10e-03    5.30e-03
residual 0.120131
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 7 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020749e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    6.07e-04    7.82e-04
   1  2.236258e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    1.06e-03    1.85e-03
   2  2.058955e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    1.01e-03    2.87e-03
   3  1.223945e-01    2.05e+01    6.80e+03   3.00e+01   1.00e+00  2.70e+05
1    9.54e-04    3.83e-03
   4  1.205858e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    1.04e-03    4.87e-03
residual 0.120586
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 8 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020772e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    5.48e-04    7.60e-04
   1  2.236155e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    1.14e-03    1.91e-03
   2  2.059116e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    9.52e-04    2.87e-03
   3  1.231485e-01    2.05e+01    6.81e+03   3.00e+01   1.00e+00  2.70e+05
1    9.50e-04    3.82e-03
   4  1.213383e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    9.95e-04    4.82e-03
residual 0.121338
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 9 -
> Trying calibrate...
```

```
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020772e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    6.81e-04    8.90e-04
   1  2.236261e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    1.16e-03    2.06e-03
   2  2.058975e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    1.07e-03    3.15e-03
   3  1.232319e-01    2.05e+01    6.81e+03   3.00e+01   1.00e+00  2.70e+05
1    1.06e-03    4.22e-03
   4  1.214222e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    1.02e-03    5.25e-03
residual 0.121422
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 10
-> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020750e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    5.49e-04    6.97e-04
   1  2.237162e+03    3.02e+10    2.96e+06   2.61e+00   1.00e+00  3.00e+04
1    9.70e-04    1.68e-03
   2  2.065169e+01    2.22e+03    5.83e+05   7.19e+02   9.91e-01  9.00e+04
1    9.50e-04    2.63e-03
   3  1.400352e-01    2.05e+01    6.78e+03   2.99e+01   1.00e+00  2.70e+05
1    9.52e-04    3.59e-03
   4  1.382403e-01    1.79e-03    1.05e+00   4.67e-01   1.00e+00  8.10e+05
1    9.54e-04    4.55e-03
residual 0.13824
Accelerometers calibration: Better calibration obtained using threshold
multiplier 6 with residual 0.120131
Misalignment Matrix
          1         -0           0
-0.00354989           1          -0
-0.00890444  -0.0213032           1
Scale Matrix
0.00241267          0          0
         0 0.00242659          0
         0          0 0.00241232
Bias Vector
33124.2
33275.2
32364.4

Accelerometers calibration: inverse scale factors:
414.478
412.102
414.538
```



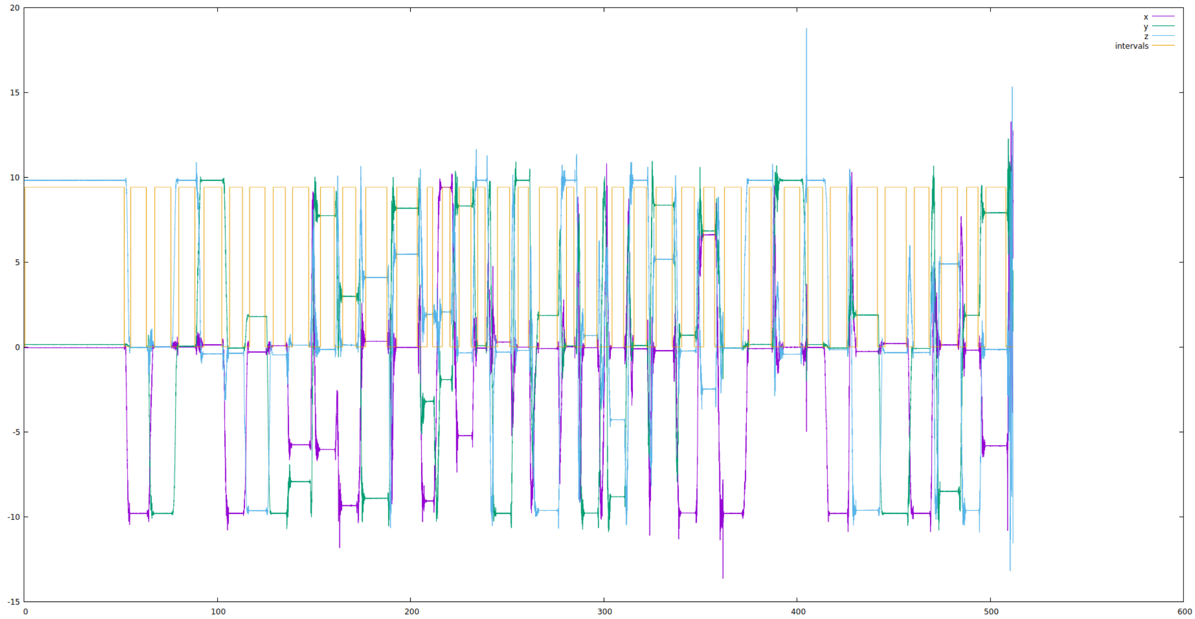
自动求导

```
Importing IMU data from the Matlab matrix file : ./test_data/xsens_acc.mat
Importing IMU data from the Matlab matrix file : ./test_data/xsens_gyro.mat
Accelerometer Calibration: Calibrating...
Accelerometers Calibration: Extracted 42 intervals using threshold multiplier 2 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.399567e+10    0.00e+00    3.17e+10   0.00e+00   0.00e+00  1.00e+04
0     5.18e-03    5.45e-03
   1  2.700191e+03    3.40e+10    5.32e+06   2.63e+00   1.00e+00  3.00e+04
1     5.77e-03    1.12e-02
   2  2.278048e+01    2.68e+03    6.95e+05   7.16e+02   9.92e-01  9.00e+04
1     5.65e-03    1.69e-02
   3  1.366645e-01    2.26e+01    7.18e+03   3.18e+01   1.00e+00  2.70e+05
1     5.63e-03    2.25e-02
   4  1.347244e-01    1.94e-03    1.18e+00   4.74e-01   1.00e+00  8.10e+05
1     5.59e-03    2.81e-02
residual 0.134724
Accelerometers Calibration: Extracted 40 intervals using threshold multiplier 3 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.189931e+10    0.00e+00    2.76e+10   0.00e+00   0.00e+00  1.00e+04
0     4.95e-03    5.11e-03
   1  2.343289e+03    3.19e+10    3.46e+06   2.60e+00   1.00e+00  3.00e+04
1     5.32e-03    1.04e-02
   2  2.118105e+01    2.32e+03    6.06e+05   7.18e+02   9.91e-01  9.00e+04
1     5.30e-03    1.58e-02
   3  1.332983e-01    2.10e+01    6.87e+03   3.17e+01   1.00e+00  2.70e+05
1     5.38e-03    2.11e-02
   4  1.313451e-01    1.95e-03    1.04e+00   4.89e-01   1.00e+00  8.10e+05
1     5.32e-03    2.65e-02
residual 0.131345
Accelerometers Calibration: Extracted 39 intervals using threshold multiplier 4 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
```

```
   0  3.112881e+10    0.00e+00    2.73e+10   0.00e+00   0.00e+00  1.00e+04
0    4.75e-03    4.92e-03
   1  2.310335e+03    3.11e+10    3.40e+06   2.61e+00   1.00e+00  3.00e+04
1    5.14e-03    1.01e-02
   2  2.117401e+01    2.29e+03    6.06e+05   7.18e+02   9.91e-01  9.00e+04
1    5.20e-03    1.53e-02
   3  1.279232e-01    2.10e+01    6.76e+03   3.00e+01   1.00e+00  2.70e+05
1    5.17e-03    2.04e-02
   4  1.261258e-01    1.80e-03    9.99e-01   4.65e-01   1.00e+00  8.10e+05
1    5.30e-03    2.58e-02
residual 0.126126
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 5 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020722e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    4.66e-03    4.82e-03
   1  2.236142e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    5.05e-03    9.88e-03
   2  2.057339e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    5.04e-03    1.49e-02
   3  1.223695e-01    2.05e+01    6.81e+03   3.01e+01   1.00e+00  2.70e+05
1    5.14e-03    2.01e-02
   4  1.205582e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    5.09e-03    2.52e-02
residual 0.120558
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 6 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020735e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    4.62e-03    4.78e-03
   1  2.236228e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    5.04e-03    9.83e-03
   2  2.057615e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    5.04e-03    1.49e-02
   3  1.219430e-01    2.05e+01    6.81e+03   3.01e+01   1.00e+00  2.70e+05
1    5.05e-03    1.99e-02
   4  1.201312e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    5.15e-03    2.51e-02
residual 0.120131
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 7 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020749e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    4.63e-03    4.79e-03
   1  2.236258e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    5.06e-03    9.85e-03
   2  2.058955e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    5.04e-03    1.49e-02
   3  1.223945e-01    2.05e+01    6.80e+03   3.00e+01   1.00e+00  2.70e+05
1    5.00e-03    1.99e-02
   4  1.205858e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    5.01e-03    2.49e-02
residual 0.120586
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 8 -
> Trying calibrate...
```

```
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020772e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    4.64e-03    4.80e-03
   1  2.236155e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    5.08e-03    9.89e-03
   2  2.059116e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    5.04e-03    1.49e-02
   3  1.231485e-01    2.05e+01    6.81e+03   3.00e+01   1.00e+00  2.70e+05
1    5.05e-03    2.00e-02
   4  1.213383e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    4.99e-03    2.50e-02
residual 0.121338
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 9 -
> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020772e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    4.59e-03    4.75e-03
   1  2.236261e+03    3.02e+10    2.96e+06   2.60e+00   1.00e+00  3.00e+04
1    5.04e-03    9.80e-03
   2  2.058975e+01    2.22e+03    5.82e+05   7.19e+02   9.91e-01  9.00e+04
1    5.04e-03    1.48e-02
   3  1.232319e-01    2.05e+01    6.81e+03   3.00e+01   1.00e+00  2.70e+05
1    4.99e-03    1.98e-02
   4  1.214222e-01    1.81e-03    1.02e+00   4.70e-01   1.00e+00  8.10e+05
1    5.07e-03    2.49e-02
residual 0.121422
Accelerometers Calibration: Extracted 38 intervals using threshold multiplier 10
-> Trying calibrate...
iter      cost      cost_change  |gradient|   |step|    tr_ratio  tr_radius
ls_iter  iter_time  total_time
   0  3.020750e+10    0.00e+00    2.59e+10   0.00e+00   0.00e+00  1.00e+04
0    4.60e-03    4.75e-03
   1  2.237162e+03    3.02e+10    2.96e+06   2.61e+00   1.00e+00  3.00e+04
1    5.00e-03    9.76e-03
   2  2.065169e+01    2.22e+03    5.83e+05   7.19e+02   9.91e-01  9.00e+04
1    5.00e-03    1.48e-02
   3  1.400352e-01    2.05e+01    6.78e+03   2.99e+01   1.00e+00  2.70e+05
1    5.04e-03    1.98e-02
   4  1.382403e-01    1.79e-03    1.05e+00   4.67e-01   1.00e+00  8.10e+05
1    5.05e-03    2.49e-02
residual 0.13824
Accelerometers calibration: Better calibration obtained using threshold
multiplier 6 with residual 0.120131
Misalignment Matrix
          1          -0           0
-0.00354989           1          -0
-0.00890444  -0.0213032           1
Scale Matrix
0.00241267          0          0
         0 0.00242659          0
         0          0 0.00241232
Bias Vector
33124.2
33275.2
32364.4
```

```
Accelerometers calibration: inverse scale factors:
414.478
412.102
414.538
```



总结和思考、疑问

1. 自动求导和解析式求导得到的标定结果一致，但是自动求导的时间稍微多那么一些。
2. 关于第5章IMU内参的标定，好像在kititi数据集中并不需要内参的标定，那么对于自己搭的一台小车，我该如何进行imu内参的标定以及标定后如何在多传感器融合中使用到内参。
3. 怎么实时进行一些内参的标定。具体来说就是怎么将第5章和第6章的一些工程代码运用到自己搭的小车上去，进行内参的标定。