Homework-05

1. 推导雅可比

(这里的符号表示和PPT相比做了些改动。。比如零偏下标应该是a)

$$\boldsymbol{\theta}^{acc}=[S_{ayx}\quad S_{azx}\quad S_{azy}\quad K_{ax}\quad K_{ay}\quad K_{az}\quad b_{ax}\quad b_{ay}\quad b_{az}]^T \tag{1}$$

$$f(\boldsymbol{\theta}^{acc})=\|g\|^2-\|a\|^2 \tag{2}$$

$$a=(I-S_a)K_a'(A-b_a) \tag{3}$$

$$\begin{aligned}J&=\frac{\partial f}{\partial\boldsymbol{\theta}}\\&=\frac{\partial(\|g\|^2-\|a\|^2)}{\partial\boldsymbol{\theta}}\\&=-\frac{\partial\|a\|^2}{\partial\boldsymbol{\theta}}\\&=-\frac{\partial\|(I-S_a)K_a'(A-b_a)\|^2}{\partial\boldsymbol{\theta}}\end{aligned} \tag{4}$$

先计算a:

$$\begin{aligned}a&=(I-S_a)K_a'(A-b_a)\\&=\begin{bmatrix}1&0&0\\-S_{ayx}&1&0\\-S_{azx}&-S_{azy}&1\end{bmatrix}\begin{bmatrix}1/K_{ax}&0&0\\0&1/K_{ay}&0\\0&0&1/K_{az}\end{bmatrix}\begin{bmatrix}A_x-b_{ax}\\A_y-b_{ay}\\A_z-b_{az}\end{bmatrix}\\&=\begin{bmatrix}1&0&0\\-S_{ayx}&1&0\\-S_{azx}&-S_{azy}&1\end{bmatrix}\begin{bmatrix}(A_x-b_{ax})/K_{ax}\\(A_y-b_{ay})/K_{ay}\\(A_z-b_{az})/K_{az}\end{bmatrix}\\&=\begin{bmatrix}(A_x-b_{ax})/K_{ax}\\(A_y-b_{ay})/K_{ay}-(A_x-b_{ax})S_{ayx}/K_{ax}\\(A_z-b_{az})/K_{az}-(A_x-b_{ax})S_{azx}/K_{ax}-(A_y-b_{ay})S_{azy}/K_{ay}\end{bmatrix}\end{aligned} \tag{5}$$

因此

$$\begin{aligned}\|a\|^2&=a_x^2+a_y^2+a_z^2\\\boldsymbol{\theta}^{acc}&=[S_{ayx}\quad S_{azx}\quad S_{azy}\quad K_{ax}\quad K_{ay}\quad K_{az}\quad b_{ax}\quad b_{ay}\quad b_{az}]^T\end{aligned} \tag{6}$$

$$
\begin{aligned}
-\frac{\partial\|a\|^2}{\partial\boldsymbol{\theta}} &= -\frac{2a_x\partial a_x}{\partial\boldsymbol{\theta}} - \frac{2a_y\partial a_y}{\partial\boldsymbol{\theta}} - \frac{2a_z\partial a_z}{\partial\boldsymbol{\theta}} \\
&= -2a_x\begin{bmatrix} 0 \\ 0 \\ 0 \\ -\frac{A_x-b_{ax}}{K_{ax}^2} \\ 0 \\ 0 \\ -\frac{1}{K_{ax}} \\ 0 \\ 0 \end{bmatrix} - 2a_y\begin{bmatrix} -\frac{A_x-b_{ax}}{K_{ax}} \\ 0 \\ 0 \\ \frac{(A_x-b_{ax})S_{ayx}}{K_{ax}^2} \\ -\frac{A_y-b_{ay}}{K_{ay}^2} \\ 0 \\ \frac{S_{ayx}}{K_{ax}} \\ -\frac{1}{K_{ay}} \end{bmatrix} - 2a_z\begin{bmatrix} 0 \\ -\frac{A_x-b_{ax}}{K_{ax}} \\ -\frac{A_y-b_{ay}}{K_{ay}} \\ \frac{(A_x-b_{ax})S_{azx}}{K_{ax}^2} \\ \frac{(A_y-b_{ay})S_{azy}}{K_{ay}^2} \\ -\frac{A_z-b_{az}}{K_{az}^2} \\ \frac{S_{azx}}{K_{ax}} \\ \frac{S_{azy}}{K_{ay}} \\ -\frac{1}{K_{az}} \end{bmatrix} \\
&= \begin{bmatrix} 2a_y\frac{A_x-b_{ax}}{K_{ax}} \\ 2a_z\frac{A_x-b_{ax}}{K_{ax}} \\ 2a_z\frac{A_y-b_{ay}}{K_{ay}} \\ 2a_x\frac{A_x-b_{ax}}{K_{ax}^2} - 2a_y\frac{(A_x-b_{ax})S_{ayx}}{K_{ax}^2} - 2a_z\frac{(A_x-b_{ax})S_{azx}}{K_{ax}^2} \\ 2a_y\frac{A_y-b_{ay}}{K_{ay}^2} - 2a_z\frac{(A_y-b_{ay})S_{azy}}{K_{ay}^2} \\ 2a_z\frac{A_z-b_{az}}{K_{az}^2} \\ 2a_x\frac{1}{K_{ax}} - 2a_z\frac{S_{azx}}{K_{ax}} \\ -2a_y\frac{S_{ayx}}{K_{ax}} - 2a_z\frac{S_{azy}}{K_{ay}} \\ 2a_y\frac{1}{K_{ay}} + 2a_z\frac{1}{K_{az}} \end{bmatrix}
\end{aligned}
\tag{7}
$$

其中

$$
\begin{aligned}
-2a_x &= -\frac{2(A_x-b_{ax})}{K_{ax}} \\
-2a_y &= -\frac{2(A_y-b_{ay})}{K_{ay}} + \frac{2(A_x-b_{ax})S_{ayx}}{K_{ax}} \\
-2a_z &= -\frac{2(A_z-b_{az})}{K_{az}} + \frac{2(A_x-b_{ax})S_{azx}}{K_{ax}} + \frac{2(A_y-b_{ay})S_{azy}}{K_{ay}}
\end{aligned}
\tag{8}
$$

$$
\tag{9}
$$

2. 完成新的内参模型下的标定, 且结果正确

修改思路

主要要看CalibratedTriad_的结构：

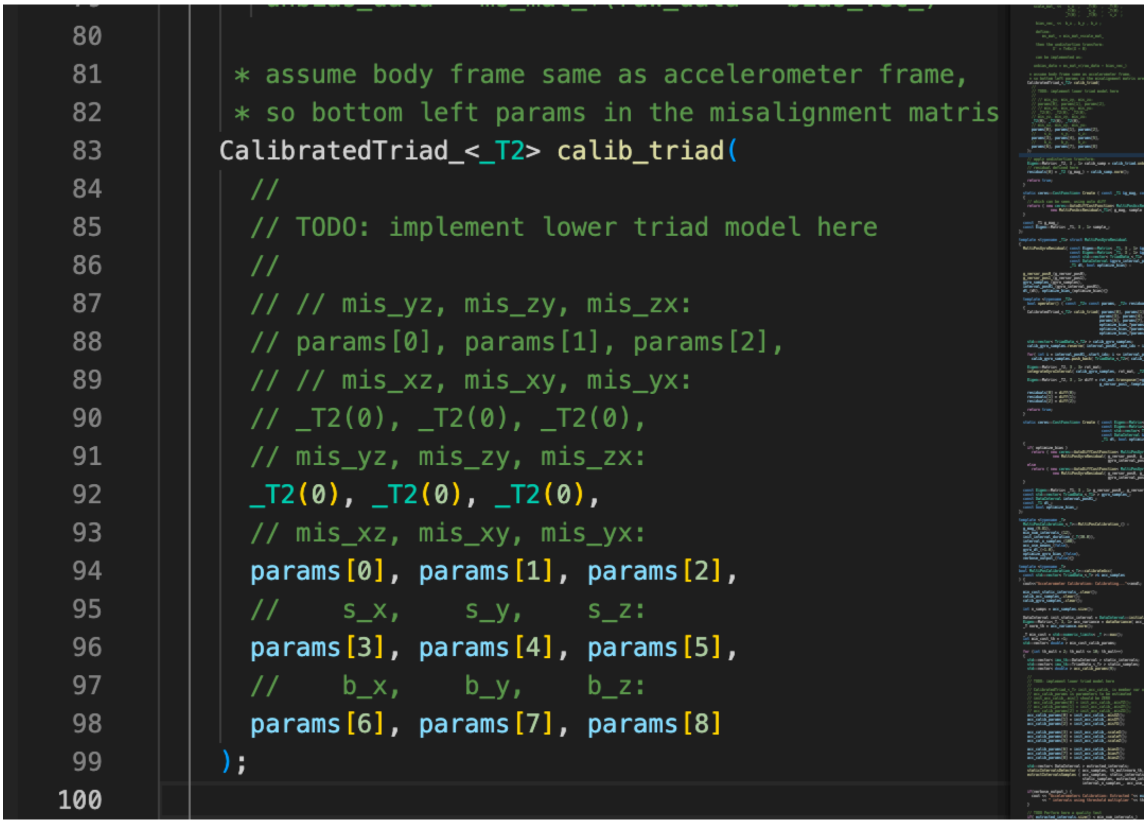
```
  */
CalibratedTriad_( const _T &mis_yz = _T(0), const _T &mis_zy = _T(0), const _T &mis_zx = _T(0),
                  const _T &mis_xz = _T(0), const _T &mis_xy = _T(0), const _T &mis_yx = _T(0),
                  const _T &s_x = _T(1),    const _T &s_y = _T(1),    const _T &s_z = _T(1),
                  const _T &b_x = _T(0),    const _T &b_y = _T(0),    const _T &b_z  = _T(0) );

~CalibratedTriad_(){};
```

```
 *          of a generic orthogonal sensor triad
 *
 * Triad model:
 *
 * -Misalignment matrix:
 *
 * general case:
 *
 *     [    1     -mis_yz   mis_zy  ]
 * T = [  mis_xz     1     -mis_zx  ]
 *     [ -mis_xy   mis_yx     1     ]
 *
 * "body" frame spacial case:
 *
 *     [  1     -mis_yz   mis_zy  ]
 * T = [  0        1     -mis_zx  ]
 *     [  0        0        1     ]
 *
 * Scale matrix:
 *
 *     [  s_x      0        0  ]
 * K = [   0      s_y       0  ]
 *     [   0       0      s_z  ]
 *
```

构造函数中，依次将参数传入，前六个为mis_yz , mis_zy , mis_zx , mis_xz, mis_xy, mis_yx。然后可以构造出如上所示的矩阵，对应ppt中的安装误差矩阵。因此，我们需要将前三个参数置0，第4-6参数设置为需要优化的误差矩阵参数。

即如下所示，在残差中设置



```
        * assume body frame same as accelerometer frame,
        * so bottom left params in the misalignment matris
    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),
      // mis_yz, mis_zy, mis_zx:
      _T2(0), _T2(0), _T2(0),
      // mis_xz, mis_xy, mis_yx:
      params[0], params[1], params[2],
      //    s_x,     s_y,     s_z:
      params[3], params[4], params[5],
      //    b_x,     b_y,     b_z:
      params[6], params[7], params[8]
    );
```

下图中，init_acc_calib_是MultiPosCalibration_中的成员变量，即初始化的calib参数，从主函数test_imu_calib.cpp可知， mis参数初始化均为0。因而我认为这里改不改，对运行没有影响。。改了之后，代码在逻辑上保持一致。

```cpp
for (int th_mult = 2; th_mult <= 10; th_mult++)
{
    std::vector< imu_tk::DataInterval > static_intervals
    std::vector< imu_tk::TriadData_<_T> > static_samples
    std::vector< double > acc_calib_params(9);

    //
    // TODO: implement lower triad model here
    //
    // CalibratedTriad_<_T> init_acc_calib_ is member va
    // acc_calib_params is parameters to be estimated
    // init_acc_calib_.mis[] should be ZERO
    // acc_calib_params[0] = init_acc_calib_.misYZ();
    // acc_calib_params[1] = init_acc_calib_.misZY();
    // acc_calib_params[2] = init_acc_calib_.misZX();
    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();
    acc_calib_params[4] = init_acc_calib_.scaleY();
    acc_calib_params[5] = init_acc_calib_.scaleZ();

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

    std::vector< DataInterval > extracted_intervals;
    staticIntervalsDetector ( acc_samples, th_mult*norm_
    extractIntervalsSamples ( acc_samples, static_interv
                              static_samples, extracted_
```

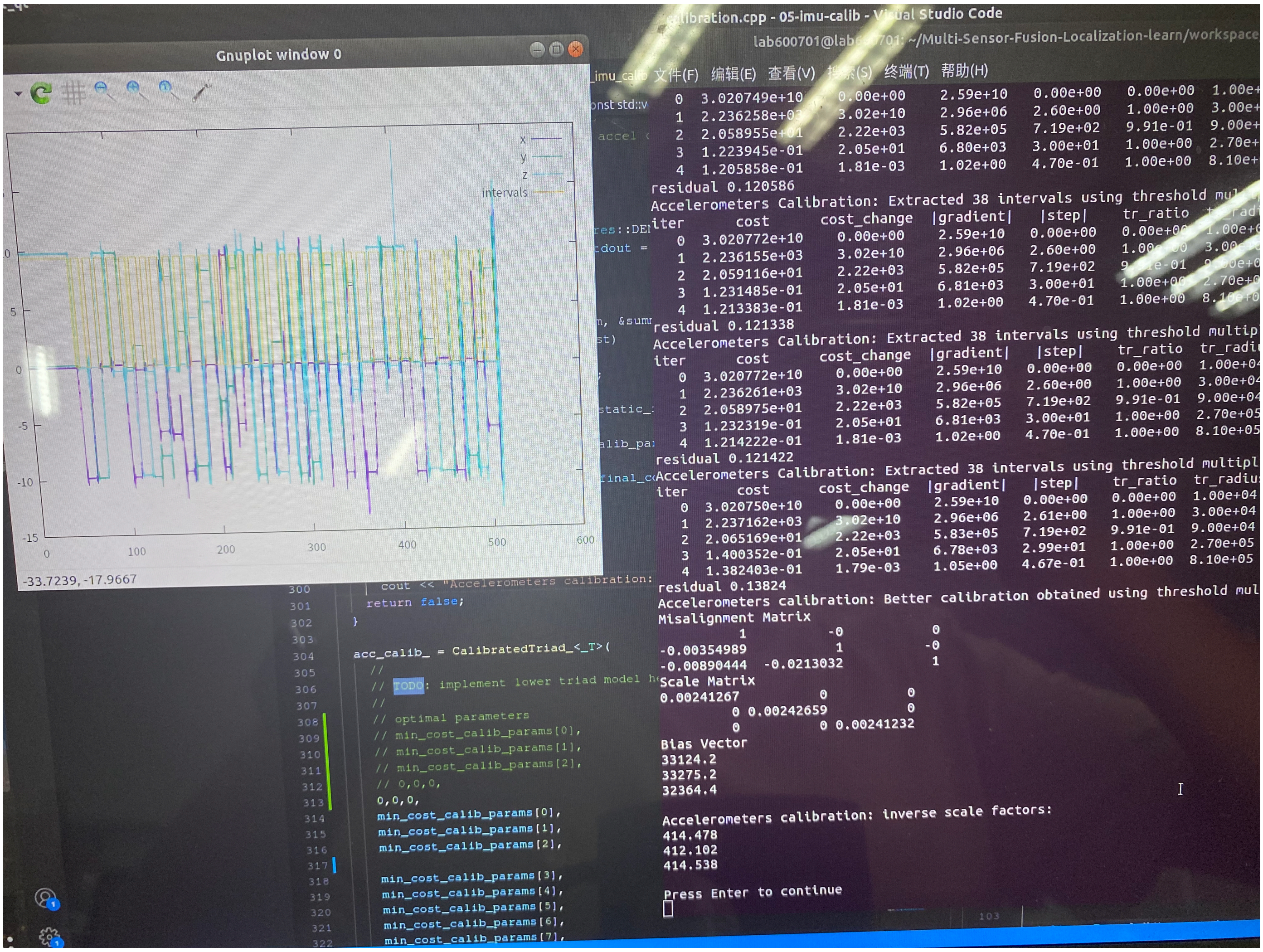
下面构造的CalibratedTriad_矩阵，应该就是最后输出到终端用的。这里也要对应前面所说的结构，前三个参数值为0，第4-6个为优化算法算出来的最优参数。

这样，最终打印输出的就是下三角矩阵。

```
acc_calib_ = CalibratedTriad_<_T>(
  //
  // TODO: implement lower triad model here
  //
  // optimal parameters
  // min_cost_calib_params[0],
  // min_cost_calib_params[1],
  // min_cost_calib_params[2],
  // 0,0,0,
  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]
);
```

成果



可以看到 misalignment matrix 变成了下三角矩阵（原先是上三角）

使用解析式求导完成标定, 且结果正确

TODO