一、雅可比推导

残差： $r_a = \|g\|^2 - \|a\|^2$

$$a = (I - S_a) K_a' (A - b_a)$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ -S_{ayx} & 1 & 0 \\ -S_{azx} & -S_{azy} & 1 \end{bmatrix} \begin{bmatrix} K'_{ax} & 0 & 0 \\ 0 & K'_{ay} & 0 \\ 0 & 0 & K'_{az} \end{bmatrix} \begin{bmatrix} A_x - b_{ax} \\ A_y - b_{ay} \\ A_z - b_{az} \end{bmatrix}$$

$$= \begin{bmatrix} K'_{ax} & 0 & 0 \\ -S_{ayx} K'_{ax} & K'_{ay} & 0 \\ -S_{azx} K'_{ax} & -S_{azy} K'_{ay} & K'_{az} \end{bmatrix} \begin{bmatrix} A_x - b_{ax} \\ A_y - b_{ay} \\ A_z - b_{az} \end{bmatrix}$$

$$= \begin{bmatrix} K'_{ax}(A_x - b_{ax}) \\ -S_{ayx} K'_{ax}(A_x - b_{ax}) + K'_{ay}(A_y - b_{ay}) \\ -S_{azx} K'_{ax}(A_x - b_{ax}) - S_{azy} K'_{ay}(A_y - b_{ay}) + K'_{az}(A_z - b_{az}) \end{bmatrix}$$

要估计参数为 $\theta = [S_{ayx}, S_{azx}, S_{azy}, K'_{ax}, K'_{ay}, K'_{az}, b_{ax}, b_{ay}, b_{az}]^T$

$$\frac{dr_a}{d\theta} = \frac{dr_a}{da} \frac{da}{d\theta} \quad , \quad \frac{dr_a}{da} = -2a^T$$

$$\frac{da}{d\theta_{123}} = \begin{bmatrix} 0 & 0 & 0 \\ -K'_{ax}(A_x - b_{ax}) & 0 & 0 \\ 0 & -K'_{ax}(A_x - b_{ax}) & -K'_{ay}(A_y - b_{ay}) \end{bmatrix}$$

$$\frac{da}{d\theta_{456}} = \begin{bmatrix} (A_x - b_{ax}) & 0 & 0 \\ -S_{ayx}(A_x - b_{ax}) & (A_y - b_{ay}) & 0 \\ -S_{azx}(A_x - b_{ax}) & -S_{azy}(A_y - b_{ay}) & (A_z - b_{az}) \end{bmatrix}$$

$$\frac{da}{d\theta_{789}} = \begin{bmatrix} -K'_{ax} & 0 & 0 \\ S_{ayx} K'_{ax} & -K'_{ay} & 0 \\ S_{azx} K'_{ax} & S_{azy} K'_{ay} & -K'_{az} \end{bmatrix}$$

$$\frac{d_{ra}}{d\theta} = -2a^T \frac{da}{d\theta}$$

$$J_1 = -2\left(-S_{ayx}k'_{ax}(A_x-b_{ax})+k'_{ay}(A_y-b_{ay})\right)\left(-k'_{ax}(A_x-b_{ax})\right)$$

$$J_2 = -2\left(-S_{azx}k'_{ax}(A_x-b_{ax})-S_{azy}k'_{ay}(A_y-b_{ay})+k'_{az}(A_z-b_{az})\right)\left(-k'_{ax}(A_x-b_{ax})\right)$$

$$J_3 = -2\left(-S_{azx}k'_{ax}(A_x-b_{ax})-S_{azy}k'_{ay}(A_y-b_{ay})+k'_{az}(A_z-b_{az})\right)\left(-k'_{ay}(A_y-b_{ay})\right)$$

$$J_4 = -2k'_{ax}(A_x-b_{ax})^2-2\left(-S_{ayx}k'_{ax}(A_x-b_{ax})+k'_{ay}(A_y-b_{ay})\right)\left(-S_{ayx}(A_x-b_{ax})\right)$$
$$-2\left(-S_{azx}k'_{ax}(A_x-b_{ax})-S_{azy}k'_{ay}(A_y-b_{ay})+k'_{az}(A_z-b_{az})\right)\left(-S_{azx}(A_x-b_{ax})\right)$$

$$J_5 = -2\left(-S_{ayx}k'_{ax}(A_x-b_{ax})+k'_{ay}(A_y-b_{ay})\right)(A_y-b_{ay})$$
$$-2\left(S_{azx}k'_{ax}(A_x-b_{ax})-S_{azy}k'_{ay}(A_y-b_{ay})+k'_{az}(A_z-b_{az})\right)\left(-S_{azy}(A_y-b_{ay})\right)$$

$$J_6 = -2\left(-S_{azx}k'_{ax}(A_x-b_{ax})-S_{azy}k'_{ay}(A_y-b_{ay})+k'_{az}(A_z-b_{az})\right)(A_z-b_{az})$$

$$J_7 = -2(k'_{ax}(A_x-b_{ax})(-k'_{ax})-2\left(-S_{ayx}k'_{ax}(A_x-b_{ax})+k'_{ay}(A_y-b_{ay})\right)(S_{ayx}k'_{ax})$$
$$-2\left(-S_{azx}k'_{ax}(A_x-b_{ax})-S_{azy}k'_{ay}(A_y-b_{ay})+k'_{az}(A_z-b_{az})\right)(S_{azx}k'_{ax})$$

$$J_8 = -2\left(-S_{ayx}k'_{ax}(A_x-b_{ax})+k'_{ay}(A_y-b_{ay})\right)(-k'_{ay})-2\left(-S_{azx}k'_{ax}(A_x-b_{ax})-S_{azy}k'_{ay}(A_y-b_{ay})+k'_{az}(A_z-b_{az})\right)$$
$$(S_{azy}k'_{ay})$$

$$J_9 = -2\left(-S_{azx}k'_{ax}(A_x-b_{ax})-S_{azy}k'_{ay}(A_y-b_{ay})+k'_{az}(A_z-b_{az})\right)(-k'_{az})$$

二、解析式求导

2.1 解析式求导代码与公式对应关系

```cpp
Eigen::Matrix<double, 3, 1> calib_samp = calib_triad.unbiasNormalize(raw_samp);

residuals[0] = g_mag_ * g_mag_ - calib_samp.transpose()*calib_samp;

if(jacobians != NULL)
{
  if(jacobians[0] != NULL)
  {
    double S1 = parameters[0][0];
    double S2 = parameters[0][1];
    double S3 = parameters[0][2];

    double K1 = parameters[0][3];
    double K2 = parameters[0][4];
    double K3 = parameters[0][5];

    double b1 = parameters[0][6];
    double b2 = parameters[0][7];
    double b3 = parameters[0][8];

    double A1 = sample_(0);
    double A2 = sample_(1);
    double A3 = sample_(2);

    Eigen::MatrixXd a(3,1);
    a << K1*(A1-b1), -S1*K1*(A1-b1) + K2*(A2-b2), -S2*K1*(A1-b1)-S3*K2*(A2-b2)+K3*(A3-b3);

    Eigen::MatrixXd da_dtheta(3,9);
    da_dtheta << 0,           0,            0,            A1-b1,        0,            0,     -K1,   0,     0,
                 -K1*(A1-b1), 0,            0,            -S1*(A1-b1),  A2-b2,        0,     S1*K1, -K2,   0,
                 0,           -K1*(A1-b1),  -K2*(A2-b2),  -S2*(A1-b1),  -S3*(A2-b2),  A3-b3, S2*K1, S3*K2, -K3;

    Eigen::Map<Eigen::Matrix<double, 1, 9, Eigen::RowMajor> > Jacob(jacobians[0]);
    Jacob.setZero();

    Jacob = -2 * calib_samp.transpose() * da_dtheta;
  }
}
```

残差： $r_a = \|g\|^2 - \|a\|^2$

$$\begin{bmatrix} k'_{ax}(A_x-b_{ax}) \\ -S_{ayx}k'_{ax}(A_x-b_{ax})+k'_{ay}(A_y-b_{ay}) \\ -S_{azx}k'_{ax}(A_x-b_{ax})-S_{azy}k'_{ay}(A_y-b_{ay})+k'_{az}(A_z-b_{az}) \end{bmatrix}$$

$$\frac{da}{d\theta_{123}} = \begin{bmatrix} 0 & 0 & 0 \\ -k'_{ax}(A_x-b_{ax}) & 0 & 0 \\ 0 & -k'_{ax}(A_x-b_{ax}) & -k'_{ay}(A_y-b_{ay}) \end{bmatrix}$$

$$\frac{da}{d\theta_{456}} = \begin{bmatrix} (A_x-b_{ax}) & 0 & 0 \\ -S_{ayx}(A_x-b_{ax}) & (A_y-b_{ay}) & 0 \\ -S_{azx}(A_x-b_{ax}) & -S_{azy}(A_y-b_{ay}) & (A_z-b_{az}) \end{bmatrix}$$

$$\frac{da}{d\theta_{789}} = \begin{bmatrix} -k'_{ax} & 0 & 0 \\ S_{ayx}k'_{ax} & -k'_{ay} & 0 \\ S_{azx}k'_{ax} & S_{azy}k'_{ay} & -k'_{az} \end{bmatrix}$$

$$\frac{dr_a}{d\theta} = -2a^T\frac{da}{d\theta}$$

2.2 解析式求导调用

```
ceres::Problem problem;
for( int i = 0; i < static_samples.size(); i++)
{
  // ceres::CostFunction* cost_function = MultiPosAccResidual<_T>::Create (
  //    g_mag_, static_samples[i].data()
  // );


  ceres::CostFunction* cost_function = new MultiPosAccResidualAnalytical<_T>(g_mag_, static_samples[i].data());


  problem.AddResidualBlock (
    cost_function,            /* error fuction */
    NULL,                     /* squared loss */
    acc_calib_params.data()   /* accel deterministic error params */
  );
}
```

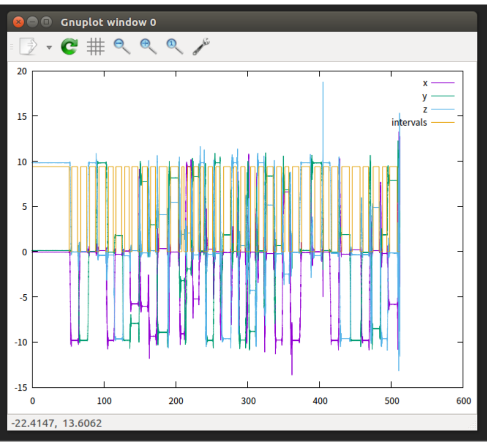
三、标定结果

3.1 未使用解析式求导

```
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    4.78e-03    5.40e-03
   1  2.237162e+03    3.02e+10    2.96e+06   2.61e+00   1.00e+00  3.00e+04        1    5.21e-03    1.06e-02
   2  2.065169e+01    2.22e+03    5.83e+05   7.19e+02   9.91e-01  9.00e+04        1    5.08e-03    1.57e-02
   3  1.400352e-01    2.05e+01    6.78e+03   2.99e+01   1.00e+00  2.70e+05        1    5.01e-03    2.07e-02
   4  1.382403e-01    1.79e-03    1.05e+00   4.67e-01   1.00e+00  8.10e+05        1    5.01e-03    2.57e-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

Press Enter to continue
```



3.2 使用解析式求导

```
  11  3.771378e+04    5.50e+03    1.08e+08   1.20e+02   1.27e-01  4.18e+08        1    1.34e-03    1.79e-02
  12  1.645018e+05   -1.27e+05    0.00e+00   7.25e+00  -3.37e+00  2.09e+08        1    4.38e-04    1.83e-02
  13  1.644821e+05   -1.27e+05    0.00e+00   7.25e+00  -3.37e+00  5.22e+07        1    4.22e-04    1.88e-02
  14  1.643639e+05   -1.27e+05    0.00e+00   7.24e+00  -3.36e+00  6.53e+06        1    4.23e-04    1.92e-02
  15  1.632711e+05   -1.26e+05    0.00e+00   7.16e+00  -3.33e+00  4.08e+05        1    4.23e-04    1.96e-02
  16  1.469687e+05   -1.09e+05    0.00e+00   6.40e+00  -2.92e+00  1.27e+04        1    4.22e-04    2.01e-02
  17  5.641510e+04   -1.87e+04    0.00e+00   6.68e+00  -1.10e+00  1.99e+02        1    4.21e-04    2.05e-02
  18  3.683780e+04    8.76e+02    6.46e+05   1.92e-01   6.47e-01  2.04e+02        1    1.31e-03    2.18e-02
  19  3.719998e+04   -3.62e+02    0.00e+00   1.96e-01  -9.19e-01  1.02e+02        1    4.34e-04    2.22e-02
  20  3.701714e+04   -1.79e+02    0.00e+00   9.86e-02  -9.07e-01  2.55e+01        1    4.27e-04    2.27e-02
  21  3.687986e+04   -4.21e+01    0.00e+00   2.48e-02  -8.47e-01  3.19e+00        1    4.25e-04    2.31e-02
  22  3.684048e+04   -2.68e+00    0.00e+00   3.10e-03  -4.29e-01  2.00e-01        1    4.23e-04    2.35e-02
residual 36837.8
Accelerometers calibration: Better calibration obtained using threshold multiplier 3 with residual 31858.6
Misalignment Matrix
          1          -0           0
  0.0618076           1          -0
-0.00865239   0.0825562           1
Scale Matrix
0.00242709          0          0
         0 0.00241928          0
         0          0 0.00242072
Bias Vector
  33119
33276.8
32368.1

Accelerometers calibration: inverse scale factors:
412.017
413.347
413.1

Press Enter to continue
```