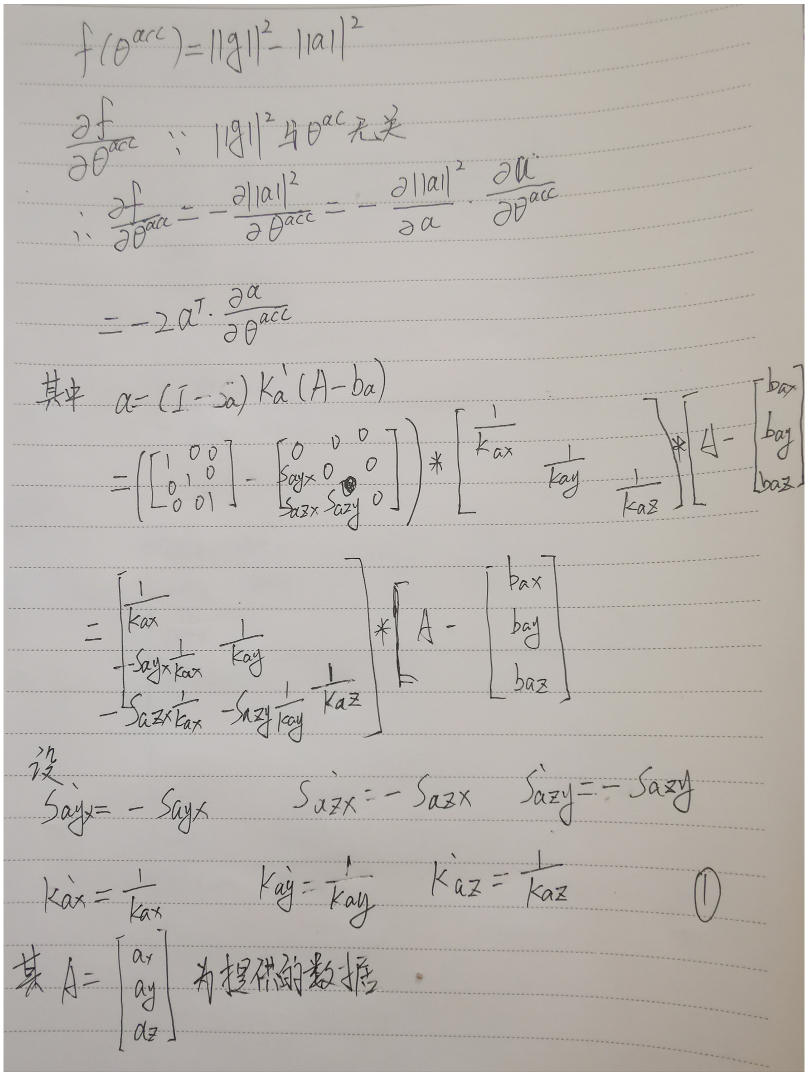
雅克比矩阵推导



下面用sympy推导的here

a矩阵

$$\left[\begin{matrix}Kax \left(Ax - bax\right)\\KaxSayx \left(Ax - bax\right) + Kay \left(Ay - bay\right)\\KaxSazx \left(Ax - bax\right) + KaySazy \left(Ay - bay\right) + Kaz \left(Az - baz\right)\end{matrix}\right]$$

对Sayx求偏导

$$\left[\begin{matrix}0\\Kax \left(Ax - bax\right)\\0\end{matrix}\right]$$

对Sazx求偏导

$$\begin{bmatrix} 0 \\ 0 \\ Kax\left(Ax - bax\right) \end{bmatrix}$$

对Sazy求偏导

$$\begin{bmatrix} 0 \\ 0 \\ Kay\left(Ay - bay\right) \end{bmatrix}$$

对Kax求偏导

$$\begin{bmatrix} Ax - bax \\ Sayx\left(Ax - bax\right) \\ Sazx\left(Ax - bax\right) \end{bmatrix}$$

对Kay求偏导

$$\begin{bmatrix} 0 \\ Ay - bay \\ Sazy\left(Ay - bay\right) \end{bmatrix}$$

对Kaz求偏导

$$\begin{bmatrix} 0 \\ 0 \\ Az - baz \end{bmatrix}$$

对bax求偏导

$$\begin{bmatrix} -Kax \\ -KaxSayx \\ -KaxSazx \end{bmatrix}$$

对bay求偏导

$$
\begin{bmatrix} 0 \\ -Kay \\ -KaySazy \end{bmatrix}
$$

对baz求偏导

$$
\begin{bmatrix} 0 \\ 0 \\ -Kaz \end{bmatrix}
$$

$$
J_f = -2a^T \cdot \begin{bmatrix} \frac{\partial a}{\partial Sayx} & \frac{\partial a}{\partial Sazx} & \frac{\partial a}{\partial Sazy} & \frac{\partial a}{\partial Kax} & \frac{\partial a}{\partial Kay} & \frac{\partial a}{\partial Kaz} & \frac{\partial a}{\partial bax} & \frac{\partial a}{\partial bay} & \frac{\partial a}{\partial baz} \end{bmatrix}
$$

代码实现

ppt中提供的公式与imu_tk论文中提供的公式不同，下面按照代码中TODO进行处理。
在calibration.h 中是这样定义的

```
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 ;
```

下面对代码进行修改:

here

```
// 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();
```

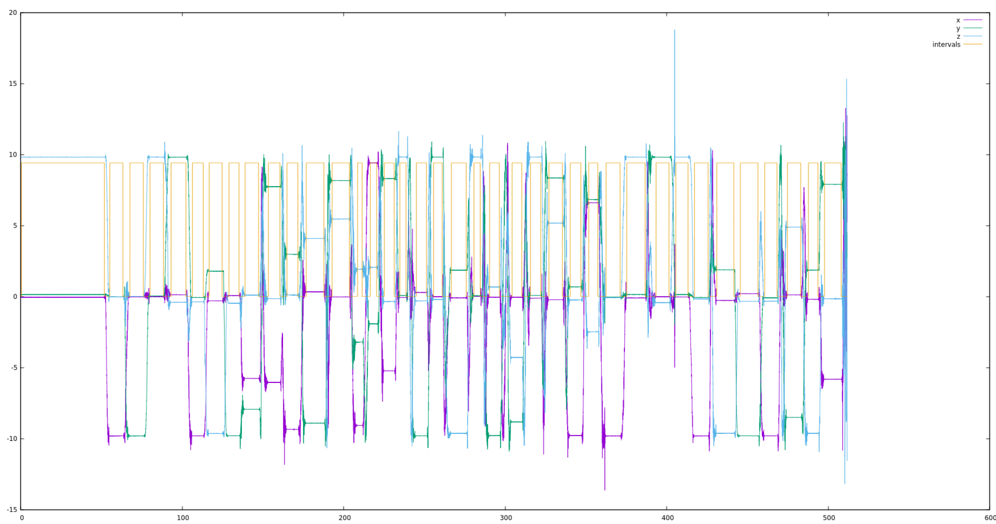
here

```
0, 0, 0,
min_cost_calib_params[0],
min_cost_calib_params[1],
min_cost_calib_params[2],
```

here

```
// mis_yz, mis_zy, mis_zx:
_T2(0), _T2(0), _T2(0),
// mis_xz, mis_xy, mis_yx:
params[0], params[1], params[2],
```

效果如下:



加速度结果

```
Accelerometers calibration: Better calibration obtained using threshold multipli
er 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
```

磁力计结果

```
Total                                     0.3064

Termination:                      CONVERGENCE (Function tolerance reached. |cost
_change|/cost: 1.098767e-11 <= 1.000000e-06)

Gyroscopes calibration: residual 0.00150696
Misalignment Matrix
         1 0.00927517 0.00990014
0.00507442          1 -0.0322229
 0.0162201 -0.0239393          1
Scale Matrix
0.000209338           0           0
          0 0.000209834           0
          0           0 0.000209664
Bias Vector
32777.1
32459.8
32511.8

Gyroscopes calibration: inverse scale factors:
4776.96
4765.68
4769.53
```