

第 11 节课习题: IMU 传感器

1. 设置 IMU 仿真代码中不同的参数, 生成 Allen 方差标定曲线

随机误差主要包括: 高斯白噪声、bias 随机游走。加速度计和陀螺仪随机误差的标定通常使用 Allan 方差法, Allan 方差法是 20 世纪 60 年代由美国国家标准局的 David Allan 提出的基于时域的分析方法。

作业给出了两个 Allen 方差标定工具:

https://github.com/gaowenliang/imu_utils

https://github.com/rpng/kalibr_allan

1.1 使用 vio_data_simulation-ros_version 生成 imu.bag

1. 可以修改 vio_data_simulation-ros_version 功能包中 param.h 头文件中加速度计和陀螺仪的高斯白噪声的标准差和 bias 随机游走的标准差, 注意这是连续时间下的值。

```

Open ▾  ⓘ  *param.h  ~/catkin_ws/src/vio_data_simulation/src  Save  ⌵  ⌵  ⌵
#ifdef IMUSIM_PARAM_H
#define IMUSIM_PARAM_H

#include <Eigen3/Eigen/Core>

class Param{
public:

    Param();

    // time
    int imu_frequency = 200;
    int cam_frequency = 30;
    double imu_timestep = 1./imu_frequency;
    double cam_timestep = 1./cam_frequency;
    double t_start = 0;
    double t_end = 3600 * 4; // 4 hours for allan

    // noise
    double gyro_bias_sigma = 0.00005;
    double acc_bias_sigma = 0.0005;

    double gyro_noise_sigma = 0.015; // rad/s * 1/sqrt(hz)
    double acc_noise_sigma = 0.019; // m/(s^2) * 1/sqrt(hz)

```

2. 新建 ros 工作环境, 将 source ~/catkin_ws/devel/setup.bash 写入 .bashrc 文件, 否则每次打开终端都要运行一次 source 命令, 挺麻烦的
3. 将功能包放到 ros 环境下编译
4. 打开 roscore
5. 执行命令 rosrunc vio_data_simulation vio_data_simulation_nod, 运行节点, 生成 imu.bag。过一会儿会在自己的 home 目录下生成 imu.bag (约 1.1GB)。

1.2 使用 imu_utils 标定随机误差

编译该功能包的过程中遇到了很多问题, 大体上和[1]中描述的问题类似:

1. 需安装依赖 sudo apt-get install libdw-dev 以及 ceres, 因为 code_imu 依赖 ceres

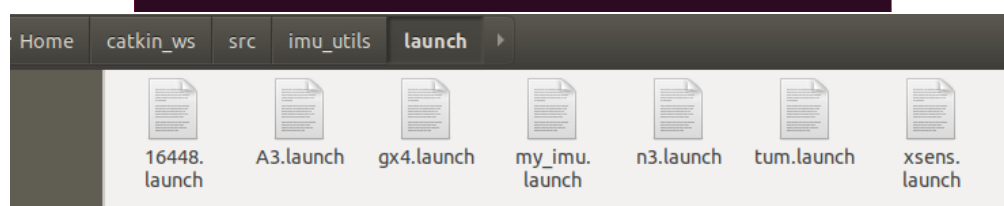
根据[1]的描述, 要先把 code_utils 放在工作空间的 src 下面, 进行编译。然后再将 imu_utils 放到 src 下面, 再编译。

2. 下载 code_utils 并编译: git clone https://github.com/gaowenliang/code_utils
编译 code_utils 时报错, 解决方法是在 code_utils 下面找到 sumpixel_test.cpp, 修改 #include "backward.hpp" 为 #include "code_utils/backward.hpp", 再编译;
3. 下载 imu_utils 并编译: git clone https://github.com/gaowenliang/imu_utils
4. 创建自己的 launch 文件 my_imu.launch, 也可以修改任意一个存在的 launch 文件, 但是注意令话题名称为 imu。这里让我白白浪费了很多时间, 就是因为没有正确输入 imu.bag 的话题名, 导致接收不到发布的数据, 无法生成计算结果的文件。可通过指令 rosbag info imu.bag 显示 bag 的话题名。

```

jindong@jindong-virtual-machine: ~
File Edit View Search Terminal Help
jindong@jindong-virtual-machine:~$ rosbag info imu.bag
path:          imu.bag
version:       2.0
duration:      3hr 59:59s (14399s)
start:         Dec 18 2021 14:19:24.88 (1639833564.88)
end:           Dec 18 2021 18:19:24.88 (1639847964.88)
size:          1.0 GB
messages:      2880001
compression:   none [1344/1344 chunks]
types:         sensor_msgs/Imu [6a62c6daae103f4ff57a132d6f95cec2]
topics:        imu 2880001 msgs ; sensor_msgs/Imu
jindong@jindong-virtual-machine:~$

```



```

Open ▾  *my_imu.launch
~/catkin_ws/src/imu_utils/launch Save ≡ ⌵ ⌵ ⌵

<launch>
  <node pkg="imu_utils" type="imu_an" name="imu_an" output="screen">
    <param name="imu_topic" type="string" value="/imu"/>
    <param name="imu_name" type="string" value="imu"/>
    <param name="data_save_path" type="string" value="$(find imu_utils)/data/" />
    <param name="max_time_min" type="int" value="120"/>
    <param name="max_cluster" type="int" value="100"/>
  </node>
</launch>

```

5. 再编译，运行 `roslaunch imu_utils my_imu.launch`，等待 imu 数据进入 topic:

```

process[imu_an-1]: started with pid [10283]
[ INFO] [1639825403.330455361]: Loaded imu_topic: /imu
[ INFO] [1639825403.332498284]: Loaded imu_name: imu
[ INFO] [1639825403.333031835]: Loaded data_save_path: /home/jindong/catkin_ws/src/imu_utils/data/
[ INFO] [1639825403.333483992]: Loaded max_time_min: 120
[ INFO] [1639825403.333940352]: Loaded max_cluster: 100
gyr x num of Cluster 100
gyr y num of Cluster 100
gyr z num of Cluster 100
acc x num of Cluster 100
acc y num of Cluster 100
acc z num of Cluster 100
wait for imu data.

```

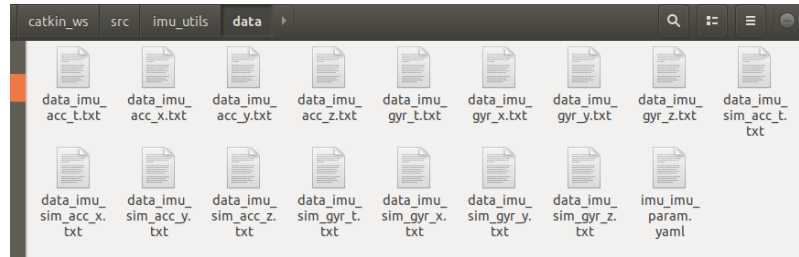
6. 打开另一个终端，回放 `imu.bag` 中的数据: `roslaunch play -r 500 ~/my_imu.bag`，等待的 `my_imu.launch` 从 topic 获得数据，并进行相应的计算,终端显示如下:

```

C -0.000105695 0.0249275 -0.00183621 0.000443017 -4.0095e-06
Bias Instability 0.0039202 m/s^2
White Noise 0.259701 m/s^2
-----
acc y
C -6.77861e-05 0.0246019 -0.00120319 0.000293872 -1.10309e-06
Bias Instability 0.0036073 m/s^2
White Noise 0.270785 m/s^2
-----
acc z
C -5.50917e-05 0.0243828 -0.000779132 0.000231998 1.63311e-06
Bias Instability 0.0038645 m/s^2
White Noise 0.265761 m/s^2
-----
[imu_an-1] process has finished cleanly
log file: /home/jindong/.ros/log/cd536f68-5fe1-11ec-bac0-000c296e552b/imu_an-1*.log
all processes on machine have died, roslaunch will exit
shutting down processing monitor...
... shutting down processing monitor complete
done
jindong@jindong-virtual-machine:~$

```

输出文件将存储到 `ros` 工作空间下的 `src/imu_utils/data/` 目录中:



标定结果位于文件 `imu_imu_param.yaml` 内，如下所示:

```

Open ▾  imu_imu_param.yaml
~/catkin_ws/src/imu_utils/data Save ≡ ⌵ ⌵ ⌵

%YAML:1.0
---
type: IMU
name: imu
Gyr:
  unit: " rad/s"
  avg-axis:
    gyr_n: 2.0988682986301069e-01
    gyr_w: 7.9196719100868356e-04
  x-axis:
    gyr_n: 2.1178508118670022e-01
    gyr_w: 9.1278735533516047e-04
  y-axis:
    gyr_n: 2.06576196233671e-01
    gyr_w: 8.4944076046996307e-04
  z-axis:
    gyr_n: 2.1129921216865510e-01
    gyr_w: 6.1367345722092726e-04
Acc:
  unit: " m/s^2"
  avg-axis:
    acc_n: 2.654157372472221e-01
    acc_w: 3.8151414279155634e-03
  x-axis:
    acc_n: 2.5970121803045970e-01
    acc_w: 3.9201991196122282e-03
  y-axis:
    acc_n: 2.7078545375988516e-01
    acc_w: 3.660726336195365e-03
  z-axis:
    acc_n: 2.6576054138382177e-01
    acc_w: 3.8644988305149247e-03

```

由于 vio_data_simulation-ros_version 功能包中 param.h 头文件中加速度计和陀螺仪的高斯白噪声的标准差和 bias 随机游走的标准差为连续时间下的标准差。我们模拟了一个连续的系统，但采集的数据确是离散的，导致 imu_utils 标定的随机误差也是离散时间下的随机误差。根据课堂上给出的公式，我们有如下关系：

$$\begin{aligned} \text{高斯白噪声: } \sigma_d &= \sigma \frac{1}{\sqrt{\Delta t}} \\ \text{Bias 随机游走的噪声: } \sigma_{bd} &= \sigma_b \sqrt{\Delta t} \end{aligned}$$

其中采样时间 Δt 可以由头文件 param.h 中的 imu_frequency = 200 Hz 得出：

$$\Delta t = \frac{1}{200} = 0.005s$$

我们将 imu_utils 计算得出的三个坐标轴上噪声标准差的平均值 avg-axis 转换到连续时间：

$$\begin{aligned} \text{陀螺仪: } \sigma^g &= 2.0988682986301069e-01 * \sqrt{0.005} = 0.01484124 \\ \sigma_b^g &= 7.9196719100868356e-04 / \sqrt{0.005} = 0.0112 \\ \text{加速度计: } \sigma^a &= 2.6541573772472221e-01 * \sqrt{0.005} = 0.01876773 \\ \sigma_b^a &= 3.8151414279155634e-03 / \sqrt{0.005} = 0.0540 \end{aligned}$$

接下来我们保持采样频率 imu_frequency = 200 Hz 不变，改变模拟的加速度计和陀螺仪的高斯白噪声的标准差和 bias 随机游走的标准差。如果将上面的模拟噪声大小视作中噪声的话，我们接下来可以模拟低噪声和高噪声的情况：

```
// noise
// 中噪声
double gyro_bias_sigma = 0.00005;
double acc_bias_sigma = 0.0005;

double gyro_noise_sigma = 0.015;    // rad/s * 1/sqrt(hz)
double acc_noise_sigma = 0.019;    // m/(s^2) * 1/sqrt(hz)

// 低噪声
/*
double gyro_bias_sigma = 1.0e-5;
double acc_bias_sigma = 0.0001;

double gyro_noise_sigma = 0.01;
double acc_noise_sigma = 0.01;
*/

// 高噪声
/*
double gyro_bias_sigma = 1.0e-4;
double acc_bias_sigma = 0.001;

double gyro_noise_sigma = 0.03;
double acc_noise_sigma = 0.03;
*/
```

然后对每种噪声情况重复之前的步骤，将标定结果转化到连续时间下的值，我们最终得到下面这张表格，来比对不同噪声大小下标定结果的好坏：

(连续时间下)	陀螺仪 σ^g	真实值	陀螺仪 σ_b^g	真实值	加速度计 σ^a	真实值	加速度计 σ_b^a	真实值
低噪声	0.0099	0.01	0.0061	1e-5	0.0098	0.01	0.0160	1e-4
中噪声	0.01484	0.015	0.0112	5e-5	0.01876773	0.019	0.0540	5e-4
高噪声	0.0299	0.03	0.0200	1e-4	0.0296	0.03	0.0966	1e-3

我们发现，用 imu_utils 标定得到的陀螺仪和加速度计的高斯白噪声的标准差与模拟中给定的值相差很小，误差在千分之一左右。但是标定的陀螺仪和加速度计的 bias 随机游走的标准差和真实值相差较大，标定值会比真实值大几百倍左右。

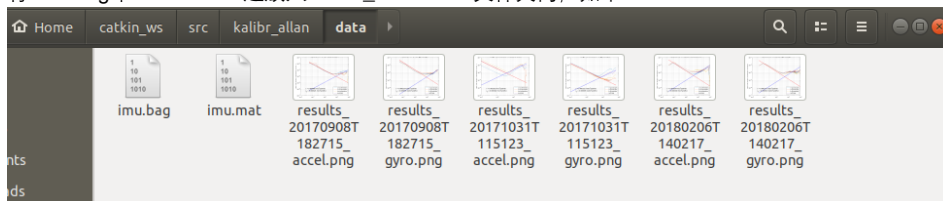
1.3 使用 kalibr_allan 标定随机误差

我们使用另一个 Allen 方差标定工具 kalibr_allan，编译该功能包的过程中遇到了比上一个标定程序更多的问题，按顺序罗列如下：

1. 在 Ubuntu18.04 上安装完 Matlab2020b 后，下载功能包 git clone https://github.com/rpng/kalibr_allan.git，在 ros 工作空间内进行编译，但由于 matlab 无法被链接导致编译失败，文献[2]中给出了解决办法。
2. 编译成功后，执行 rosrund bagconvert bagconvert imu.bag imu
(注意 topic 名称应与 imu.bag 相同，为 imu。这里又是一个坑，若 topic 名称与 imu.bag 不一致，虽然也会生成 imu.mat 文件，但是生成的文件为空，无法在后续步骤中进行绘图。)成功会在 home 目录下生成 imu.mat 文件(约 138MB)。终端显示如下：

```
jindong@jindong-virtual-machine: ~  
File Edit View Search Terminal Help  
jindong@jindong-virtual-machine:~$ rosrund bagconvert bagconvert imu.bag imu  
[ INFO] [1639860451.528597898]: Starting up  
[ INFO] [1639860453.711876957]: BAG Path is: imu.bag  
[ INFO] [1639860453.711935371]: MAT Path is: imu.mat  
[ INFO] [1639860453.711948570]: Reading in rosbag file...  
[ INFO] [1639860457.503189330]: Done processing bag  
[ INFO] [1639860462.868734745]: Done processing IMU data  
jindong@jindong-virtual-machine:~$
```

将 imu.bag 和 imu.mat 一起放入 kalibr_allan/data 文件夹内，如下：



3. 打开 matlab，修改 kalibr_allan/matlab/SCRIPT_allan_matparallel.m 文件内的路径如下：

```
SCRIPT_allan_matparallel.m* +  
1 %% Initialization  
2 - close all  
3 - clear all  
4  
5 % Read in our toolboxes  
6 - addpath('functions/allan_v3')  
7  
8 % Our bag information  
9  
10 %mat_path = '../data/imu_mtig700.mat';  
11 %mat_path = '../data/imu_tango.mat';  
12 mat_path = '../data/imu.mat';  
13  
14 % IMU information (todo: move this to the yaml file)  
15 %update_rate = 400;  
16 %update_rate = 100;  
17 update_rate = 200;  
18
```

并选择 update_rate = 200，与 imu 的频率一致。

运行 SCRIPT_allan_matparallel.m，发现错误，因为没有安装 Parallel Computing Toolbox：

```
SCRIPT_allan_matparallel.m* +  
1 %% Initialization  
2 - close all  
3 - clear all  
4  
5 % Read in our toolboxes  
6 - addpath('functions/allan_v3')  
7  
8 % Our bag information  
9 %mat_path = '../data/imu_mtig700.mat';  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382  
383  
384  
385  
386  
387  
388  
389  
390  
391  
392  
393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640  
641  
642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687  
688  
689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742  
743  
744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898  
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071  
1072  
1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156  
1157  
1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251  
1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334  
1335  
1336  
1337  
1338  
1339  
1340  
1341  
1342  
1343  
1344  
1345  
1346  
1347  
1348  
1349  
1350  
1351  
1352  
1353  
1354  
1355  
1356  
1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432  
1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446  
1447  
1448  
1449  
1450  
1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460  
1461  
1462  
1463  
1464  
1465  
1466  
1467  
1468  
1469  
1470  
1471  
1472  
1473  
1474  
1475  
1476  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488  
1489  
1490  
1491  
1492  
1493  
1494  
1495  
1496  
1497  
1498  
1499  
1500  
1501  
1502  
1503  
1504  
1505  
1506  
1507  
1508  
1509  
1510  
1511  
1512  
1513  
1514  
1515  
1516  
1517  
1518  
1519  
1520  
1521  
1522  
1523  
1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665  
1666  
1667  
1668  
1669  
1670  
1671  
1672  
1673  
1674  
1675  
1676  
1677  
1678  
1679  
1680  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808  
1809  
1810  
1811  
1812  
1813  
1814  
1815  
1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972  
1973  
1974  
1975  
1976  
1977  
1978  
1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2
```

```

Command Window
New to MATLAB? See resources for Getting Started.
>> SCRIPT_allan_matparallel
opening the mat file.
loading timeseries.
imu frequency of 200.00.
sample period of 0.00500.
calculating allan deviation.
Elapsed time is 4078.141500 seconds.
saving to: results_20211218T230644.mat
done saving!
fx >>

```

在 kalibr_allan/data 目录下生成 result 文件。

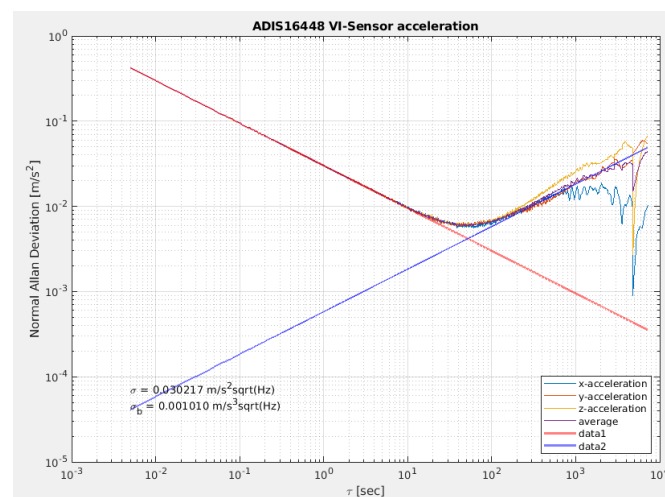
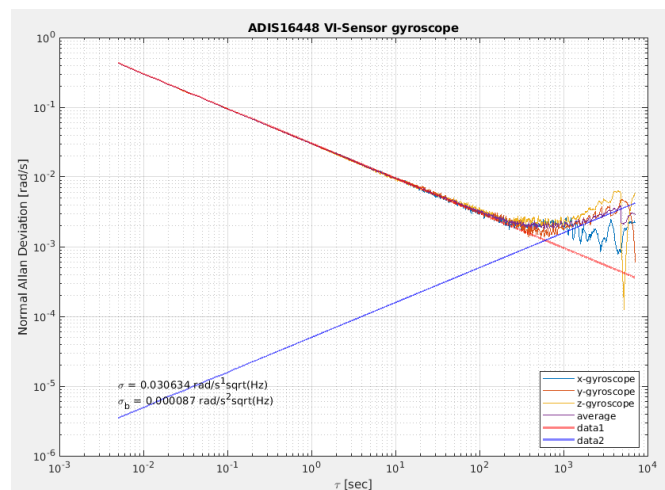
4. 将生成的 result 文件的路径添加到 kalibr_allan/matlab/SCRIPT_process_results.m 文件内，并运行该文件：

```

src ▶ kalibr_allan ▶ matlab ▶
Editor - /home/jindong/catkin_ws/src/kalibr_allan/matlab/SCRIPT_process_results.m*
SCRIPT_allan_matparallel.m  SCRIPT_process_results.m*  +
1  %% Initialization
2  close all
3  clear all
4
5  % Read in our toolboxes
6  addpath('functions')
7  addpath('functions/allan_v3')
8
9  % Our bag information
10 %titlestr = 'XSENS MTi-G-710';
11 %mat_path = '../data/bags/results_20170908T182715.mat';
12
13 %titlestr = 'Tango Yellowstone #1';
14 %mat_path = '../data/bags/results_20171031T115123.mat';
15
16 titlestr = 'ADIS16448 VI-Sensor';
17 mat_path = '../data/results_20211218T230644.mat';
18
19 % Load the mat file (should load "data_imu" matrix)
20 fprintf('=> opening the mat file.\n')
21 load(mat_path);
22

```

结果绘制出 allan 方差曲线：



我们既可以从曲线中读取各随机误差的方差值，也可以从 matlab 的 command 窗口下读到数据结果：

```
>> SCRIPT_process_results
=> opening the mat file.
=> plotting accelerometer.
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL. For
more information, click here.
tau = 1.00 | tauid1 = 1089
h_fit1 slope = -0.5000 | y-intercept = -3.4994
h_fit2 slope = 0.5000 | y-intercept = -7.4451
tau = 2.99 | tauid2 = 1201
=> plotting gyroscope.
tau = 1.00 | tauid1 = 1089
h_fit1 slope = -0.5000 | y-intercept = -3.4857
h_fit2 slope = 0.5000 | y-intercept = -9.8938
tau = 2.99 | tauid2 = 1201
=> final results
accelerometer_noise_density = 0.03021669
accelerometer_random_walk   = 0.00101031
gyroscope_noise_density     = 0.03063356
gyroscope_random_walk       = 0.00008730
fx >>
```

kalibr_allan 标定工具考虑了从 imu 采集到的离散数据的频率，生成的方差为连续时间下的方差。

我们使用的 imu.bag 是在前面提到的高噪声的情况下生成的，标定值和真实值的比对如下：

(连续时间下)	陀螺仪 σ^g	真实值	陀螺仪 σ_b^g	真实值	加速度计 σ^a	真实值	加速度计 σ_b^a	真实值
低噪声	-	0.01	-	1e-5	-	0.01	-	1e-4
中噪声	-	0.015	-	5e-5	-	0.019	-	5e-4
高噪声	0.03063356	0.03	8.730e-5	1e-4	0.03021669	0.03	1.01031e-3	1e-3

由于运行一次 SCRIPT_allan_matparallel.m 的时间过长，所以未对低噪声和中噪声两种情况进行标定，不过标定效果应该和高噪声的情况相差不大。

我们发现，该 kalibr_allan 标定工具标定的陀螺仪和加速度计的高斯白噪声的标准差和 bias 随机游走的标准差和真实模拟值相差不多，比起使用 imu_utils 标定的随机误差准确很多。

2. 将 IMU 仿真代码中欧拉积分替换成中值积分

文件夹 vio_data_simulation 的源文件 imu.cpp 中函数 MotionData IMU::MotionModel(double t) 对整个运动进行了建模。函数 void IMU::testImu(std::string src, std::string dist) 中读取生成的 imu 数据并用 imu 动力学模型对数据进行计算，最后保存 imu 积分以后的轨迹。其中的积分使用了欧拉积分，我们可以将其替换成中值积分。代码如下：

```
Eigen::Quaterniond Qwb_last(init_Rwb_);

for (int i = 1; i < imudata.size(); ++i) {

    MotionData imupose = imudata[i];
    MotionData imupose_last = imudata[i-1];

    //delta_q = [1, 1/2 * thetax, 1/2 * thetax_y, 1/2 * thetax_z]
    Eigen::Quaterniond dq;

    // Eigen::Vector3d dtheta_half = imupose.imu_gyro * dt / 2.0; // 欧拉积分
    Eigen::Vector3d dtheta_half = (imupose_last.imu_gyro * dt + imupose.imu_gyro * dt) / 2.0 / 2.0; // 中值积分

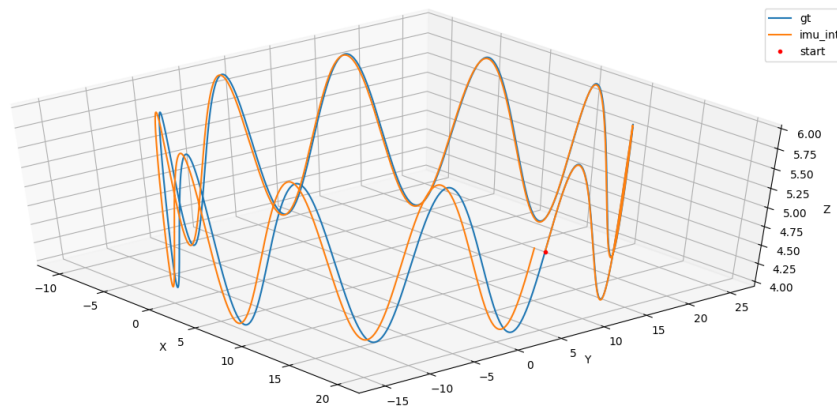
    dq.w() = 1;
    dq.x() = dtheta_half.x();
    dq.y() = dtheta_half.y();
    dq.z() = dtheta_half.z();
    dq.normalize();

    /// imu 动力学模型 欧拉积分
    /*
    Eigen::Vector3d acc_w = Qwb * (imupose.imu_acc) + gw; // aw = Rwb * ( acc_body - acc_bias ) + gw
    Qwb = Qwb * dq;
    Pwb = Pwb + Vw * dt + 0.5 * dt * dt * acc_w;
    Vw = Vw + acc_w * dt;
    */

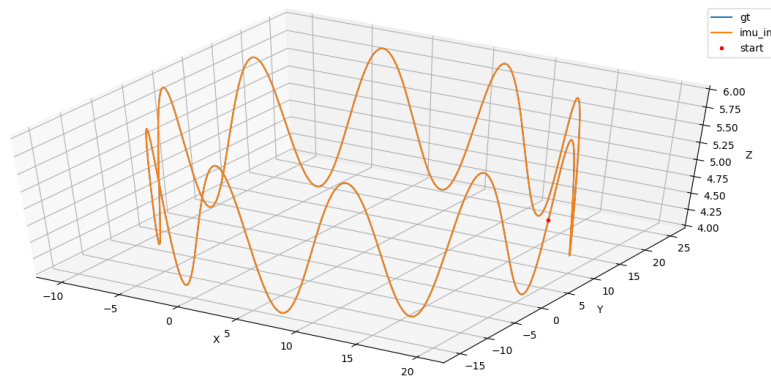
    /// 中值积分
    Qwb = Qwb_last * dq;
    Eigen::Vector3d acc_w = 0.5 * (Qwb_last * imupose_last.imu_acc + gw + Qwb * imupose.imu_acc + gw);
    Pwb = Pwb + Vw * dt + 0.5 * dt * dt * acc_w;
    Vw = Vw + acc_w * dt;
    Qwb_last = Qwb;
```

我们创建新的变量 Qwb_last 和 imupose_last 来保存上一个时刻的旋转四元数和 imu 位姿，用于中值法的计算。中值法与欧拉法的不同就在于中值法计算加速度和角速度时使用的是这个时刻和上一个时刻的平均值。使用欧拉法和中值法得到的积分轨迹如下：

欧拉法:



中值法:



我们发现欧拉法得到的积分结果和真实轨迹之间存在差异，而中值法得到的轨迹和真实轨迹很好地重合到了一块儿，所以中值法的积分结果更准确。

(因为无法看到蓝色轨迹，一开始怀疑中值法计算有误，于是在 draw_trajcory.py 中对其中一条轨迹乘上一个缩放因子再进行画图，就能看到两条曲线的差异了，从而证明是两条轨迹重合到了一起，导致只能看到一条轨迹)

编译以及绘制轨迹使用命令如下:

```
cd vio_data_simulation
./build.sh
bin/data_gen
cd bin
./data_gen           //会在 bin 文件夹内生成 imu 数据
cd python_tool
./draw_trajcory.py
```


3. 阅读从已有轨迹生成 imu 数据的论文[4]，撰写总结推导

对于任意一条曲线的拟合，如果我们用插值法(如牛顿法或拉格朗日法)拟合 n 个离散点的话，拟合的曲线的阶数就是 $n-1$ 。改变其中任意一个离散点的位置，或者其中一个离散点特别不准的话，就会对整体的拟合曲线造成很大的影响，因此这种方法不是特别鲁棒，且只适用于离散点较少的情况。如果我们使用近似法来拟合曲线(如最小二乘法)，虽然该拟合方法较为鲁棒，拟合函数阶数可以不是很高，且适合离散点较多的情况，但是拟合效果很大程度上取决于使用的拟合曲线的形式和阶数，换言之我们还是需要得到系统的模型才行。而如果我们使用样条插值 (spline interpolation)，则可以避免使用较高阶数的曲线形式，同时每个拟合点也会考虑附近点的情况，且每根样条之间足够平滑，就可以将所有样条连接起来，形成一条拟合曲线，来拟合所有采样点。而该论文就是采用了 B 样条来拟合任意一条轨迹，从而计算出相应的 imu 数据。

现将论文的主要推导过程总结如下：

1. B 样条介绍

首先全文引入了 B 样条这个概念， $k-1$ 阶 B 样条利用其基函数可以表达为：

$$p(t) = \sum_{i=0}^n p_i B_{i,k}(t)$$

其中 $p_i \in R^n$ 为 $t_i, i \in [0, \dots, n]$ 时刻的控制点， $B_{i,k}$ 为 B 样条基函数。上式可以改写成 cumulative basis function 形式：

$$p(t) = p_0 \tilde{B}_{0,k}(t) + \sum_{i=1}^n (p_i - p_{i-1}) \tilde{B}_{i,k}(t)$$

其中 $\tilde{B}_{i,k}$ 为 B 样条的 cumulative 基函数。通过使用对数和指数映射我们可以利用上式来描述 SE3 下控制点的差异：

$$\Omega_i = \log(T_{w,i-1}^{-1} T_{w,i}) \in \mathfrak{se}(3)$$

从而来刻画 t 时刻 SE3 上的轨迹：

$$T_{w,s}(t) = \exp(\tilde{B}_{0,k}(t) \log(T_{w,0})) \prod_{i=1}^n \exp(\tilde{B}_{i,k}(t) \Omega_i)$$

其中 $T_{w,s}(t) \in SE(3)$ 是样条在 t 时刻的位姿， $T_{w,i} \in SE(3)$ 是控制位姿(control poses)。我们用下标 w 来强调 t 时刻的位姿和控制位姿是在世界坐标系下给出的。

2. 累计立体 B 样条

然后论文介绍了这里主要使用的是 cumulative cubic B-Splines ($k=4$)，可以翻译为累计立体 B 样条，使用 4 个控制点来控制一个 B 样条。更准确的，对于任意一个 $t \in [t_i, t_{i+1})$ 时刻的位姿，我们使用 $[t_{i-1}, t_i, t_{i+1}, t_{i+2}]$ 的控制点来描述。更进一步我们可以将其表示为 $s(t) = (t - t_0)/\Delta t$ ，使任意控制点时间点 t_i 可以转化为归一化下的时间点 $s_i \in [0, 1, \dots, n]$ 。对于 $s_i \leq s(t) < s_{i+1}$ 的时刻点我们可以定义 $u(t) = s(t) - s_i$ 。使用这种新的时间形式，我们可以重写 B 样条的 cumulative 基函数 $\tilde{B}(u)$ ，并对其求时间导数 $\dot{\tilde{B}}(u), \ddot{\tilde{B}}(u)$ 为：

$$\tilde{B}(u) = C \begin{bmatrix} 1 \\ u \\ u^2 \\ u^3 \end{bmatrix}, \quad \dot{\tilde{B}}(u) = \frac{1}{\Delta t} C \begin{bmatrix} 0 \\ 1 \\ 2u \\ 3u^2 \end{bmatrix}, \quad \ddot{\tilde{B}}(u) = \frac{1}{\Delta t^2} C \begin{bmatrix} 0 \\ 0 \\ 2 \\ 6u \end{bmatrix}, \quad C = \frac{1}{6} \begin{bmatrix} 6 & 0 & 0 & 0 \\ 5 & 3 & -3 & 1 \\ 1 & 3 & 3 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

每根样条代表轨迹的位姿可以定义为：

$$T_{w,s}(u) = T_{w,i-1} \prod_{j=1}^3 \exp(\tilde{B}(u)_j \Omega_{i+j})$$

这里 i 代表了 B 样条的时间区间 $t \in [t_i, t_{i+1})$ ， $\tilde{B}(u)_j$ 代表 $\tilde{B}(u)$ 从 0 开始的第 j 个控制点。对于任意 u 我们定义 $\tilde{B}(u)_0 = 1$ ，我们可以表示出对应的 t 时刻的位姿的一阶和二阶导数：

$$\begin{aligned} \dot{T}_{w,s}(u) &= T_{w,i-1} \left(\dot{A}_0 A_1 A_2 + A_0 \dot{A}_1 A_2 + A_0 A_1 \dot{A}_2 \right), \\ \ddot{T}_{w,s}(u) &= T_{w,i-1} \left(\ddot{A}_0 A_1 A_2 + A_0 \ddot{A}_1 A_2 + A_0 A_1 \ddot{A}_2 + 2(\dot{A}_0 \dot{A}_1 A_2 + \dot{A}_0 A_1 \dot{A}_2 + A_0 \dot{A}_1 \dot{A}_2) \right), \\ A_j &= \exp(\Omega_{i+j} \tilde{B}(u)_j), \quad \dot{A}_j = A_j \Omega_{i+j} \dot{\tilde{B}}(u)_j, \\ \ddot{A}_j &= \dot{A}_j \Omega_{i+j} \dot{\tilde{B}}(u)_j + A_j \Omega_{i+j} \ddot{\tilde{B}}(u)_j \end{aligned}$$

3. VI 模型参数化

我们可以将以上 B 样条产生位姿及其导数的方法应用到 VI 系统当中。我们使用逆深度法将 a 帧下任意路标点的相机坐标系下的位置转化为 b 帧下对应的图像坐标：

$$p_b = \mathcal{W}(p_a; T_{b,a}, \rho) = \pi \left([K_b | 0] T_{b,a} [K_a^{-1} \begin{bmatrix} p_a \\ 1 \end{bmatrix}; \rho \right)$$

其中 $\pi(\mathbf{P}) = \frac{1}{f_z} [\mathbf{P}_0, \mathbf{P}_1]^T$ 为归一化投影函数, $\mathbf{K}_a, \mathbf{K}_b \in \mathbb{R}^{3 \times 3}$ 为相机在 a 帧和 b 帧的内参矩阵。

之前推导的累积 B 样条允许我们计算出位姿的一阶和二阶导数, 我们应用它来计算加速度计和陀螺仪的测量误差模型:

$$\begin{aligned}\text{Gyro}(u) &= \mathbf{R}_{w,s}^T(u) \cdot \dot{\mathbf{R}}_{w,s}(u) + \text{bias}, \\ \text{Accel}(u) &= \mathbf{R}_{w,s}^T(u) \cdot (\ddot{\mathbf{s}}_w(u) + \mathbf{g}_w) + \text{bias},\end{aligned}$$

前面我们已经算出位姿的一阶二阶导数了, 这里 $\dot{\mathbf{R}}_{w,s}$ and $\ddot{\mathbf{s}}_w$ 只不过是 $\dot{\mathbf{T}}_{w,s}$ and $\ddot{\mathbf{T}}_{w,s}$ 的 sub-matrices。这里 \mathbf{g}_w 是世界坐标系下的重力加速度。

4. 最小化测量和预测模型

现在我们已经有了 VI 系统模型, 我们可以求出样条函数和相机内参数举证, 通过最小化以下 objective function:

$$\begin{aligned}E(\theta) &= \sum_{\mathbf{p}_m} \left(\hat{\mathbf{p}}_m - \mathcal{W}(\mathbf{p}_r; \mathbf{T}_{c,s} \mathbf{T}_{w,s}(u_m)^{-1} \mathbf{T}_{w,s}(u_r) \mathbf{T}_{s,c}, \rho) \right)_{\Sigma_p}^2 + \\ &\quad \sum_{\hat{\omega}_m} \left(\hat{\omega}_m - \text{Gyro}(u_m) \right)_{\Sigma_\omega}^2 + \sum_{\hat{\mathbf{a}}_m} \left(\hat{\mathbf{a}}_m - \text{Accel}(u_m) \right)_{\Sigma_a}^2,\end{aligned}$$

在连续时间下, 我们同时最小化了投影误差和惯性误差(inertial error), 基于我们在固定长度的滑动窗口内得到的所有测量值, 包括路标点投影位置 $\hat{\mathbf{p}}_m$, 惯性观测量 $\hat{\omega}_m$ and $\hat{\mathbf{a}}_m$ 。

参考文献:

[1] 使用 imu_utils 对 IMU 进行标定

<https://blog.csdn.net/hltt3838/article/details/111135343>

[2] 安装 Kalibr, 在执行 catkin_make 时出现错误 The following variables are used in this project, but they are set to NO

https://blog.csdn.net/weixin_43247057/article/details/101211669

[3] License Manager Error -9 when run as administrator but not normally

https://blog.csdn.net/Strive_For_Future/article/details/108119510

[4] 2013 BMVC, Steven Lovegrove ,Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras.