

基础作业，必做

- ① 设置 IMU 仿真代码中的不同的参数，生成 Allen 方差标定曲线。
allan 方差工具：
https://github.com/gaowenliang/imu_utils
https://github.com/rpng/kalibr_allan
...
- ② 将 IMU 仿真代码中的欧拉积分替换成中值积分。

作业一：

先声明一个区别（不知道为什么）：

kalibr 认为随机游走在 $1/2$ 斜率曲线的 $t=3$ 处取得

utils 认为随机游走在曲线的最低点取得

我们在仿真功能包 `vio_sim` 中设置陀螺仪和加速度计的参数如下：

```
double gyro_bias_sigma = 0.00005;  
double acc_bias_sigma = 0.0005;  
//double gyro_bias_sigma = 1.0e-5;  
//double acc_bias_sigma = 0.0001;  
double gyro_noise_sigma = 0.015;    // rad/s  
double acc_noise_sigma = 0.019;    // m/(s^2)
```

kalibr_allan 的 **matlab** 代码存在问题，采用 **imu_utils** 方法

得到 IMU 仿真数据集 **imu.bag** 之后，可以一边回放数据集：

```
rosbag play -r 200 imu.bag
```

一边运行程序：

```
roslaunch imu_utils my.launch
```

my.launch 如下：

```
<launch>  
  <node pkg="imu_utils" type="imu_an" name="imu_an"  
output="screen">  
    <param name="imu_topic" type="string" value=  
"/imu"/>
```

```

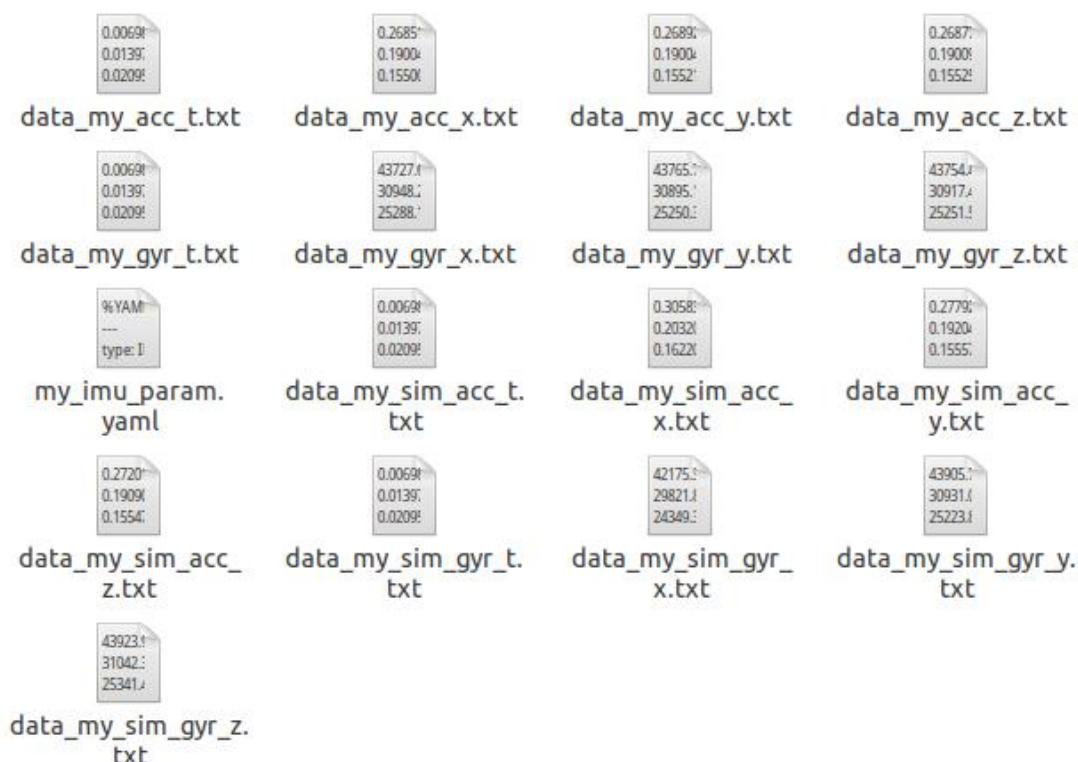
    <param name="imu_name" type="string" value=
"my"/>
    <param name="data_save_path" type="string"
value= "$(find imu_utils)/data/" />
    <param name="max_time_min" type="int" value=
"120"/>
    <param name="max_cluster" type="int" value=
"100"/>
  </node>
</launch>

```

imu_name 是 imu 的名字，只需要不与其他参与 allan 方差标定的 IMU 名字相同即可

imu_topic 是 imu 话题名称，我们的 bag 包回放的 imu 话题为 /imu，这里我们订阅它。

这样，在 **imu_utils** 功能包下的 **data** 目录下生成一些文件，为：



注意其中的 my_imu_param.yaml 内容，他是 IMU allan 方差标定的结果，如下：

```

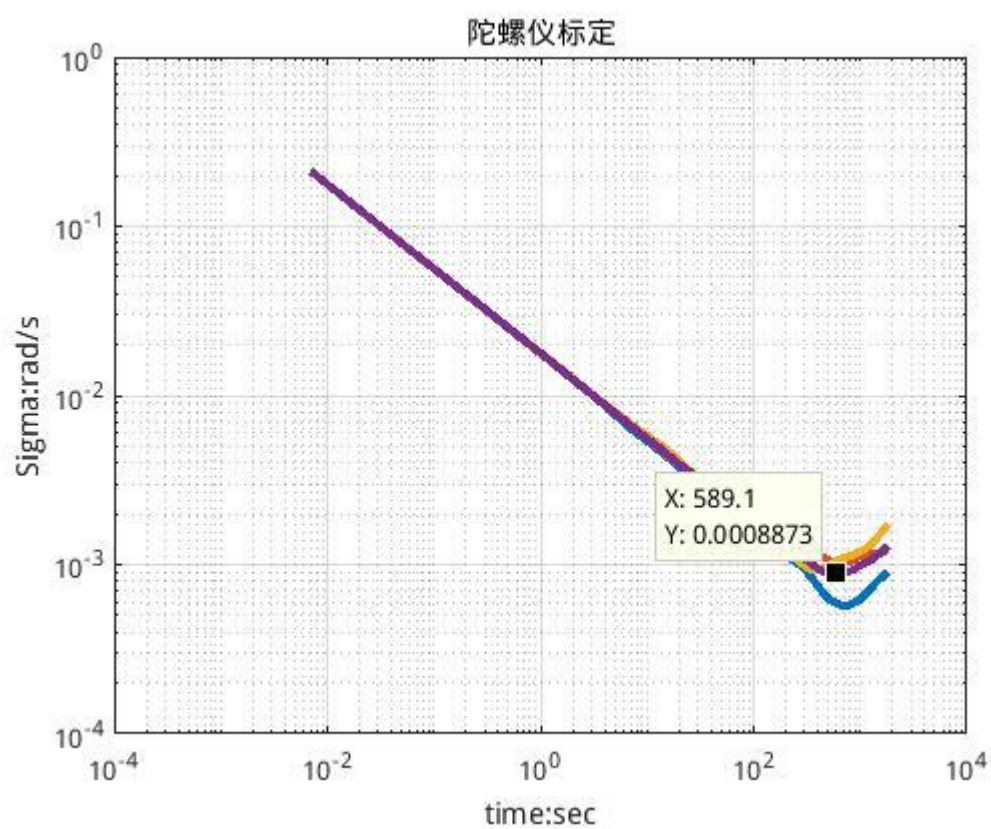
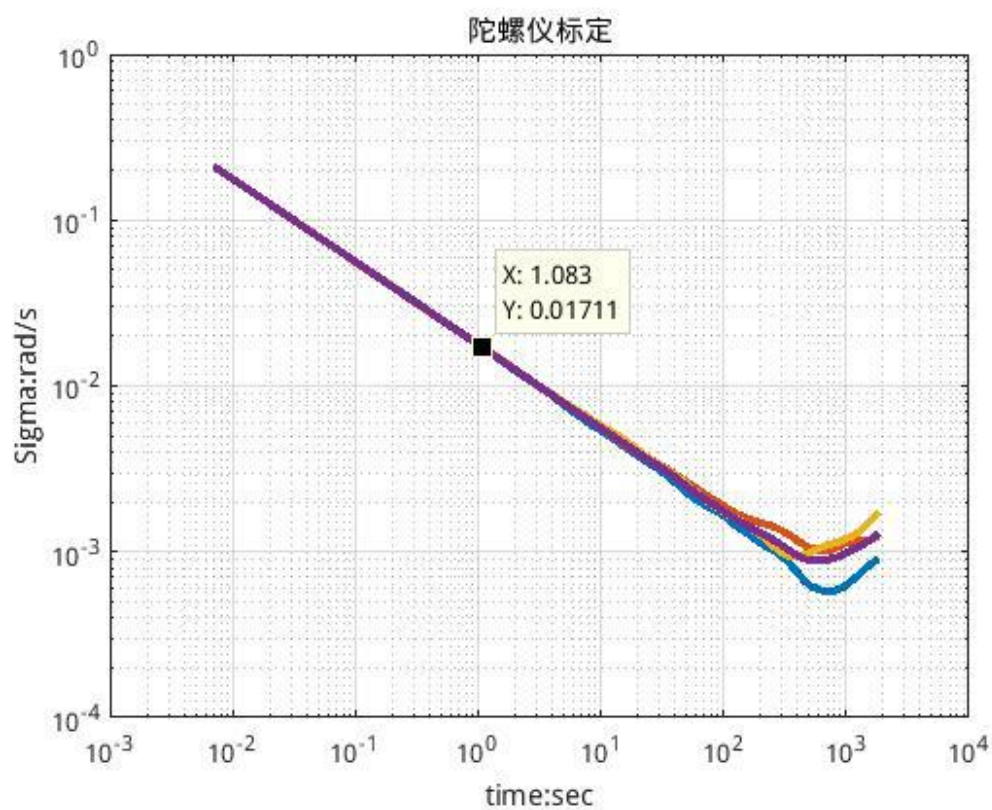
%YAML:1.0
---
```

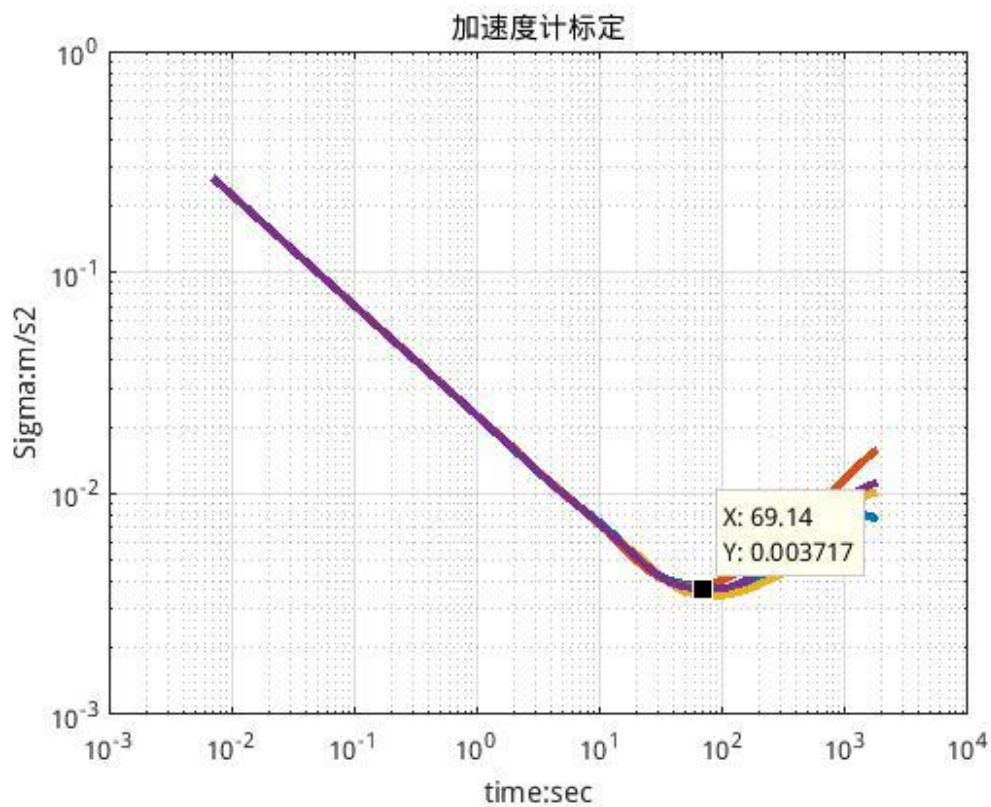
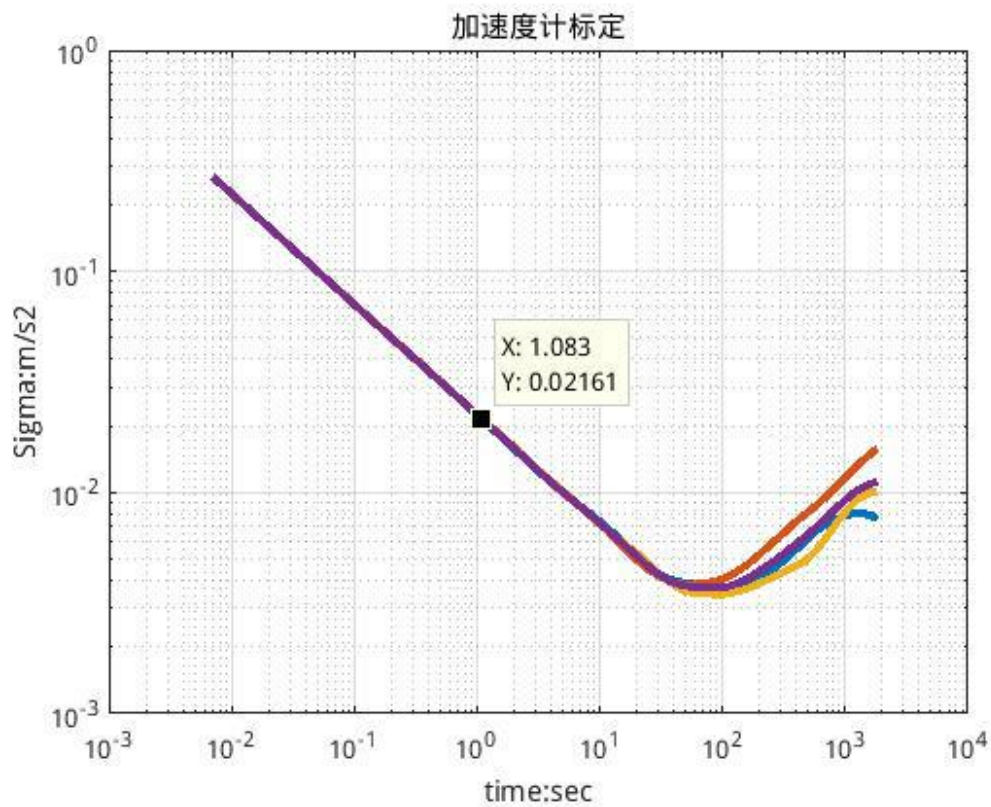
```
type: IMU
name: my
Gyr:
  unit: " rad/s"
  avg-axis:
    gyr_n: 2.0953877119978270e-01 ( 实际应该是-02 )
    gyr_w: 9.1066956888183543e-04
  x-axis:
    gyr_n: 2.0444874965775275e-01
    gyr_w: 6.8468434206845686e-04
  y-axis:
    gyr_n: 2.1145997810515862e-01
    gyr_w: 1.0622659267394978e-03
  z-axis:
    gyr_n: 2.1270758583643673e-01
    gyr_w: 9.8505843783755140e-04
Acc:
  unit: " m/s^2"
  avg-axis:
    acc_n: 2.6721909383502923e-01 ( 实际应该是-02 )
    acc_w: 3.6157677976775119e-03
  x-axis:
    acc_n: 2.6808046458545720e-01
    acc_w: 3.4756079524495848e-03
  y-axis:
    acc_n: 2.6524487270897440e-01
    acc_w: 3.9822337038829952e-03
  z-axis:
    acc_n: 2.6833194421065626e-01
    acc_w: 3.3894617366999565e-03
```

注意：

这里的白噪声结果和 **allan** 曲线得到的结果会相差一个数量级，而随机游走的结果与 **allan** 方差曲线大致相同（暂时还不知道为何）

下面利用 **matlab** 工具显示它们的 **allan** 方差曲线，如下：





结果可以看出：不论是陀螺仪还是加速度计，白噪声方差与真实方差相差不大，但随机游走误差相差一个数量级左右。

作业二：

改欧拉积分为中值积分：

```
for (int i = 1; i < imudata.size(); ++i) {
    Eigen::Quaterniond dq;
    Eigen::Vector3d dtheta_half =
(imudata[i-1].imu_gyro +
imudata[i].imu_gyro)*dt/4.0;
    dq.w() = 1;
    dq.x() = dtheta_half.x();
    dq.y() = dtheta_half.y();
    dq.z() = dtheta_half.z();

    Eigen::Quaterniond QwbBefore = Qwb;
    Qwb = Qwb * dq;
    Eigen::Vector3d acc_w =
(QwbBefore*(imudata[i-1].imu_acc) +
Qwb*(imudata[i].imu_acc) + 2*gw)/2.0;
    Vw = Vw + acc_w * dt;
    Pwb = Pwb + Vw * dt + 0.5 * dt * dt * acc_w;
```

plot 轨迹如下：

